# The S.T.G Device
# Smart Tabletop Gaming

**Senior Design**
**Fall 2019 (Group 10)**


Richard Almario: .................................. Computer Engineering
Hubert Barrantes: ................................ Computer Engineering
Jose Castano: ...................................... Electrical Engineering
Coraima Orochena: .............................. Electrical Engineering

# Table of Contents

# Table of Figures

# Table of Tables

# 1.0 Executive Summary

The Smart Tabletop gaming (S.T.G.) device is a two-part device that can bridge players and their group when they can't meet up physically. There are a number of virtual tabletops (VTT)'s, but this version utilizes a physical board and a virtual board at the same time. They will be able to communicate their actions to their group without the overhead of correcting their actions, like with a VTT, but the group of people who are not using a VTT will able to see these actions take place on the main board. To help express the intent of the removed player's action, a LED array is employed to convey actions from the web app, while a sensor array on the board can pick up actions from a pointer tool.

Regular players utilizing the web app VTT will have their movement actions translated to the board, and a XY plotter physically navigating their game piece to the destination point. This XY plotter is situated beneath the board, with a magnetic piston setup on the actuating device. Each game piece will have a small magnet attached to their bottom, which the XY plotter will be able manipulate. To detect the game pieces on the main board and sync those pieces to the web VTT, the sensor array located under board tracks each grid square, reference the known piece locations, while any new signatures are transmitted to the database for the game piece position coordinates. The web app then picks up this update via a web socket event and then updates it's the VTT user interface. Other actions that can be transmitted are pointing out specific positions and performing special effects via the LED array, or adding/removing their game pieces from the board. Features on the web application will allow a player to track basic information on their game piece, such as health points and armor class.

This board is not just for the players of the group but also can be used by the Game Master. This role also goes by the title Dungeon Master (DM), and how that relates to the board is that the game that will be demonstrated on the board is Dungeon's and Dragons. The DM will be able to add or remove any game pieces on the board, designate locations in which a game piece cannot navigate, as well as see information on any game piece that is on the web app VTT.

The framework of the web app allows for scalability and improvements to the overall functionality of the S.T.G. device. With all changes happening server-side, multiple boards could by synced to a single game session, while more features can be added to the web app. For this project alone, they are sticking to a smaller feature set and building in the foundation that will allow for further development.

# 2.0 Project Description

In this section, the project was generally described in terms of motivation, goals, objectives to follow and describing the main equipment essential to the project which includes XY plotter, controller, web application and server.

## 2.1 Project Motivation/Goal

When it comes to Tabletop games, one thing that keeps people from trying out this genre of gaming is they want to play with a group they are familiar with. Another would be that one or more people would like to join into an ongoing session, but they are in faraway locations. Also, most people would want to have a tangible feeling to their game and do not like to use a virtual tabletop. This project could provide a solution to a variety of people with different needs.

In this case, the game is featuring is Dungeons and Dragons, a fantasy role-playing game based on medieval myth. In a regular game, it will include game pieces, dice, paper, pencil and the people who are attending. The most important part for the game to proceed is the presence of people who want to play the game. This project will make that requirement an easier goal to accomplish and will allow people to attend the game session with a physical presence that you don't get through a virtual tabletop.

There have been various ways to include absent people onto a game session, such as using web cameras and having a mic to pick up the group's conversation. Most of these methods are effective up to a point but usually run into the difficulty of conveying the actions from the absent player to the gameboard. The Dungeon Master must interpret these displaced actions correctly and devote a certain amount of attention to perform these actions onto the tabletop. This slows down the session and can lead to some frustration. To fix this issue, the web application communicates to the tabletop all the various actions that a person will want to communicate to the group that is physically at the tabletop.

This implementation removes the middleman who must listen in and perform the actions while keeping the game session flowing smoothly. This middleman being the DM, who's time is best suited to guiding the campaign rather than micromanagement.

## 2.2 Project Objectives

The goal of this project is the following: to be able to move the game pieces using a magnetic XY plotter, to create a web application that will be synced to the physical tabletop, and for the player using the application, to be able to express their actions to that tabletop in various ways. These different actions can be spell effects, movement actions, and pointing out specific locations that player is interested in.

So far, the size of the board is one of the most critical steps for this project since it was leading to an x number of magnetic sensors. The size was reasonably big so that the game pieces can move freely and make the game interesting, the grid space was roughly 10in by 13in. Next, an XY plotter was built and it was used to move the game pieces around the board underneath the table. The programmable LED's was set up as well to light up the board.

### 2.2.1 The Web application and Server

HTML and JavaScript was utilized to create a web application that can communicate to the Device Controller. Various API's such as Node JS or a MVC framework was being used to access a server such as MySQL or MongoDB database. This application allowed the participant to click on the screen and it sent the data to the XY plotter controller to move a game piece physically on the board. It also continuously monitors all the game objects on the board by referencing their positional data on the server. On the application, it displayed a login Username, Health points, Armor class, Speed, and a feature that can send an attack to the enemy. Also, the enemy sent the attack to the participant which can deplete the Health level. On the server, the information stored contain all the unique ID's of the game objects was currently populating the tabletop device.

### 2.2.2 Controller

The controller communicates with the web server and receives information for any game object actions. It communicated with the XY plotter for moving the game objects around the board. One feature that was implemented was an edit mode combined with a magnetic pen that can be used to design structures of castles, trees, etc. to create an atmosphere to the game. These designs were utilized to create obstacles in the grid. Let's say the participant would like to move his or her game piece across a wall in which the XY plotter will not move the game piece through that wall. However, the XY plotter can find another path to move the game piece around the wall. In other words, the XY plotter can find its nearest and most efficient path for the game piece to move to its designated location.

If there is no valid path to navigate the game piece to its destination or if the distance was greater than the speed value, then the board can notify to the application that the piece can't move and there was a visible indication on the board that the game piece can't move. Manual movement of game pieces on the board was monitored by the sensors and the controller picks up this action. When an object was removed, the controller references what object was in that position and when that object is placed on a new position on the board, the controller updates that object's positional data on the server. This game objects position is updated on the web application due to its monitoring capabilities.

### 2.2.3 XY Plotter

The functionality of the XY plotter is to execute physical moments in X and Y motion once it was receiving software commands from the controller. The assembly was using the main components such as step motors, micro servo, and belts in order to move the game pieces on a transparent surface. The plan was to connect the stepper motors and the LEDs to ULN2003 chip since the chip contains multiple relays in order to receive multiple information and data from the controller and activation of LEDs during gameplay. However, it was convenient to use the 74HC595 and the 74HC165 shift registers for the LED connection only instead of ULN2003. Since the original functionality of the XY plotter was drawing, configuration was necessary in order to hold not just a pencil, but an electromagnet with the maximum force of 210N which was essential for the tabletop gaming. The game pieces were going to be glued with a magnetic material so that the electromagnet was able to grab and move the piece to the designated location. The XY plotter will be located on the bottom of the surface to move the game piece.

### 2.2.4 Sensors

The sensors were needed to detect the user's movement and sense what's occupied on the board. Previously, the Hall effect sensors were the primary choice since the XY plotter had a magnetic sensor, therefore the sensors must be magnetic, so they connect. Hall effect sensors were magnetic, they were also either latched or non-latched. The sensors need to be latched because the sensors need to turn on by a north magnetic pole and turned off by a south magnetic pole, which is what latched sensors do. The resistors are needed to be kOhm range (most likely 10 kOhm) to connect the sensors with the voltage supply, output, and ground. They will also need FET multiplexers/demultiplexers to differentiate between sensors and to input and output what the program sets. Since tabletop board was 10 in. by 13 in. grid, they will need 256 sensors for the whole matrix. Due to that, the reed-switches were used for the project which was the connection in series with a diode and the reed-switch.

### 2.2.5 LEDs

LEDs was essential for the tabletop gaming and a feature to create a visual effect for D&D players especially for beginners. Their primary function was to light up the designated location for the game piece to go to. When installing the LEDs, LED drivers were used to behave like a power supply. However, it was ineffective to use every LED driver to every LED. Instead, an electronic chip called LM3402 was able to solve this problem. However, the LM3402 had limited functionalities which led to a decision to choose an 74HC595 and the 74HC165 shifting registers. This chip direct and redirect individual LEDs to light up once the game pieces start to move. However, the connectivity was complex and a tedious process to make every LED to work. If lighting up LEDs individually would be a problem, then using the LED strips would be the alternative. In this method, the LEDs were to light up simultaneously and creating a path when moving the game piece.

## 2.3 Requirements, Standards, & Constraints

When building the project, there will be requirements for the team members to implement, constraints to mitigate, and standards to follow. Requirements were important because the requirements will tell the group what they need or is necessary for the project to work. Constraints were important because they let the group know their limits for any requirements found if something must be a certain way. Standards were important to show what is compatible, reliable, and productive for any of the materials needed in the project.

### 2.3.1 Requirements

Web applications were important for the project because this is part of how the software was going to work and how it was going to be programmed. The user interface was important because it monitored what was going on in the game. The application was important because it monitored and updated the game from the server. In Table 1 below, this shows the requirements for the web application for the project.

**Table 1: Web Application Requirements**

| Name | Purpose |
|---|---|
| **User Interface** | – Display the current state of the board on the physical device.<br>– When moving game object from one position to another, will update the server with object new coordinates.<br>– If a move is successful, display board will update with new object’s position |
| **Application** | – Maintain a constant feed from server for any update from device<br>– Will read from the server if an attempted coordinate update is successful |

The controller was important to the project because it was the microcontroller that was used to program everything in the board. It was needed for the game board, the sensor matrix, and the XY plotter. It was also used to monitor for the game, and to program the lighting of each individual LED. Table 2 below shows everything that the controller was going to do and that was required for the project.

| Table 2: Controller Requirements | |
|---|---|
| **Name** | **Purpose** |
| **Monitors the Gameboard** | – The Controller monitored the gameboard for any input<br>– The Controller kept track of positions for game objects |
| **Receive input from Sensor Array** | – Was continuously receiving input from the magnetic sensor array and kept the track of how |
| **Output pathing to XY Plotter** | – The Controller received a move action from the database and created a proper path for the game object to navigate to its destination |
| **Checks updates from the Database** | – The Controller was continuously checking updates from the database for any changes within the game objects<br>– Database changed the game object coordinates translated to game object movement |
| **Outputs LED pattern to Sensor Array** | – The Controller was taking a pathing list of coordinates and turned on the destination coordinate LED with a specific color |

For the project, having a database was important because it kept track and organized all the different objects in the game. It saved the game object data and the user's data. The table below shows the requirements needed for the database requirements.

| Table 3: Database Requirements | |
|---|---|
| **Name** | **Purpose** |
| **Saves Game object data** | – Each game object had a database entry for specific fields such as a Unique ID, Name, Health Points, Armor Class, Speed number, and Position Coordinates |
| **Saves User data** | – Each user had a database entry containing basic information as the email, unique ID per user, and password. |

The XY plotter was important for the project because with this device was how the games pieces was going to move physically. The XY plotter required an open-end timing belt, stepper motors, magnet holder, micro servo, and ULN2003A. The table below shows details of each requirement needed for the XY plotter.

| Table 4: XY Plotter Requirements | |
|---|---|
| **Name** | **Purpose** |
| **Open-end Timing Belt** | – To move the magnetic holder X and Y direction |
| **Stepper Motors** | – To provide smooth rotation |
| **Magnetic Holder** | – Able to hold a magnetic sensor |
| **Micro Servo** | – To put the magnetic piece against the transparent surface underneath |
| **ULN2003A** | – Essential for Stepper motors' functionality using multiple arrays for movement and activation of LEDs |

The sensors were important because they were responsible for detecting the XY plotter's magnet to move the game pieces around. A sensor matrix was going to be made to be placed under each square on the board. The group was deciding whether to use hall-effect sensors or reed switches, so they were both included in the table below, but only one of them was used. The table below shows the rest of the sensor requirements.

| Table 5: Sensor Requirements | |
|---|---|
| **Name** | **Purpose** |
| **Latching Hall effect sensors** | – To detect the user's move and to sense what's occupied on the board |
| **10 kOhm resistors** | – To connect the sensors to each other with the voltage supply, output, and ground |
| **FET Multiplexers/Demultiplexers** | – To differentiate between sensors and to input and output what the program sets |
| **Reed Switches** | – To detect the user's move and to sense what's occupied on the board |

The LED's were needed to light up the board to make it look nice, and to light up paths in the game. The LED's were used mostly for aesthetic purposes, but will light up the paths in the game or light up on a game piece when assigned. The table below shows what is required to make the LED matrix.

| Table 6: LED Requirements | |
|---|---|
| **Name** | **Purpose** |
| **LED Strips** | – To light up the board |
| **LED Driver** | – To supply power to the LED's |
| **MAX16802B** | – To selectively power the LEDs to light up the designated location for the game piece to move |

### 2.3.2 Standards

For this project, the group must abide by certain standards to make sure that there is compatibility, reliability, and productivity to the development pipeline. The following Table 7 to Table 9 define standards that pertain to the project.

| Table 7: Electrical Standards | |
|---|---|
| **Task** | **Standard** |
| **Connecting the Stepper Motors** | – Must be less than 110 V |
| **Connecting Arduino MEGA 2560 with MOSFET and Diode** | – The circuitry must be clean and avoid overheating and overload |
| **XY Plotter Configuration** | – The weight should be safely distributed |
| **LED Configuration** | – Can only be powered up to 5V. |

| Table 8: Communication Protocol Standards | |
|---|---|
| **Task** | **Standard** |
| **HTTP & HTTPS** | – Safely transfer data over the web |
| | – Data encryption |

| Table 9: Coding Standards | |
|---|---|
| **Task** | **Standard** |
| **Code Conventions** | – Proper formatting.<br>– Programming principles<br>– Code quality |

### 2.3.3 Constraints

In doing the project, they may be possible realistic constraints appeared in doing specific tasks and procedure. These constraints needed to be mentioned and accounted for before the beginning the project in order to mitigate risks and get the possible best outcome.

In the beginning of the software development, types of constraints were mentioned to be aware while doing the project. In the first stage, coding was taking place and this process developed a few constraints. The lines of code must be in quality which then required a long period of time and prototyping may inflict project's budget.

| Table 10: Web Application Constraints | |
|---|---|
| **Task** | **Potential problem** |
| **Coding** | – It took time to develop a fully functional code to test connectivity from application to device |
| **Debugging** | – Fixed any bugs that may come up while programming the web interface. |
| **Time** | – It took a lengthy amount of time to program the application interface that is needed for the web application |
| **Quality assurance** | – May not detect bugs at an early stage which results in longer testing time |
| **Prototype** | – Costs extra money, excessive focus on one part of the product may result in other parts neglected |

In assembling the XY plotter, the main components can create certain constraints that might sabotage the project. This information is explained on Table 11.

| Table 11: XY Plotter Constraints | |
|---|---|
| **Task** | **Potential problem** |
| **Open-end Timing Belt** | – It may not calibrate well when moving the game piece |
| **Stepper Motors** | – It may not receive coding commands from the software and not able to execute the lines of code well. ` |
| **Magnet Holder** | – It may not hold the magnetic sensor very well. |
| **Micro Servo** | – It may not touch the transparent surface with the magnetic piece in order to move the game piece. |
| **LD** | – It may lead to complex connectivity for the step motors and LEDs |

When building the sensor, few constraints must be considered including the components such as the resistors to have a magnitude of 10Kohms and the type of sensors must be connected to the XY plotter with a magnet. This information can be mentioned on Table 12.

| Table 12: Sensor Constraints | |
|---|---|
| **Name** | Purpose |
| **Sensors** | – They must be latched Hall effect sensors because they need to be able to connect with the XY plotter which has a magnet |
| **Resistors** | – They must be kOhm resistors |

When it comes to the LED's the constraints that come with it are for them to be able to be individually programmed. Also, because they are LED's they need to be powered by LED drivers. The Table 13 below shows the minimal constraints for the LED's.

| Table 13: LED Constraints | |
|---|---|
| **Name** | **Potential problem** |
| **LED Strips** | – LED strips must be individually addressable, so it can be programmed by the controller. |
| **LED Power Supply** | – Will need to be powered by LED drivers and MAX16802B |

Social constraints are important to consider in the project because it is important to know how the project will influence society. The Table 14 below shows any social constraints that can come from this project.

| Table 14: Social Constraints | |
|---|---|
| **Name** | **Potential problem** |
| **Game Violence** | – Parents may not like the setting or premise behind playing D&D |

Political constraints that can be considered in the project are not too many because it's just based off of a simple game on Table 15

| Table 15: Political Constraints | |
|---|---|
| **Name** | **Potential problem** |
| **Dungeons and Dragon Age Rating** | – The game may not be suitable for younger |

Assembling the project may create harmful effects for the environment. This information appears to be on Table 16.

| Table 16: Environmental Constraints | |
|---|---|
| **Name** | **Potential problem** |
| **Game Piece Materials** | – The game pieces are made of plastic |
| **Server Environmental Impact** | – Server uptime helps contribute to extra electrical usage<br>– Increased electrical usage indirectly contributes to extra greenhouse emissions |

Setting up the project can results in more increased expenditure than previously anticipated. This is information is explained on Table 17.

| Table 17: Economic Constraints | |
|---|---|
| **Name** | **Potential problem** |
| **Pricing for Materials** | – Various materials have to be bought premade in order to save time. |

The XY plotter's components are manufactured by different companies which can cause adversaries. This information is explained on Table 18.

| Table 18: Sustainability Constraints | |
|---|---|
| **Name** | **Potential problem** |
| **XY Plotter maintenance** | – Plotter's parts are made of various materials that would require the tabletop disassembly |

Safety is the main concern when building this project. It is imperative to be aware and apply these safety practices to mitigate risks. This is information is explained on Table 19.

| Table 19: Health & Safety Constraints | |
|---|---|
| **Name** | **Potential problem** |
| **Moving Parts hazard** | – Lots of moving parts within the assembly could catch loose objects |

## 2.4 Project Budget and Financing

After researching a few materials for each type of device, budget expectations are listed below within Table 20 below. The biggest expense is expected to be the XY

Plotter, since creating a plotter from base materials is a huge undertaking by itself. Design and testing of a custom XY plotter would consume a significant amount of development time, and as such, the group have chosen to buy that section of the device.

| Table 20: Budget Expectations | | |
|---|---|---|
| Unit | Quantity | Cost |
| **XY Plotter** | 1 | $150-$400 |
| **Sensors** | 256 | $150 |
| **LEDs** | 256 (16x16) | $50-$70 |
| **Controller** | 1 | $40-$70 |
| **Server** | 1 | $100 |
| **Power Source** | 1 | $80 |
| **Magnet** | Variable | $2-$20 |
| **Total** | | $820 |

## 2.5 Milestones for project

The project's milestones was set to motivate and keep track of the current progress of the Senior Design Project. This represents documentation, software and hardware tasks, research, and possible goals that the group can achieve by a certain timeline.

### 2.5.1 Senior Design 1 Milestone

Senior Design 1 milestone consists the documentation process. Here is **Error! Reference source not found.** that consists main dates to complete the report:

| Table 21: Senior Design 1 Milestones | | |
|---|---|---|
| **Week** | Date | Activity |
| **2** | May 21st | Project for senior design chosen |
| **3** | May 28th | Begin of research, analyze all components and task required to accomplish project |
| **5** | June 15th | Begin writing design document |
| **8** | July 2nd | Check-up, at least 50% of paper complete |
| **11** | July 25th | Complete final design document |
| **12** | August 1st | Submit final design paper, order parts to begin assembling project |

### 2.5.2 Senior Design 2 Milestone

After the documentation process was completed, using the information from the report would be a helpful guide to fulfill project's parameters.

Here is Table 22, which consisting all the main dates along with each specific task:

| Table 22: Senior Design 2 Milestones | | |
|---|---|---|
| **Week** | Date | Activity |
| **1** | August 27th | Start assembling project |
| **3** | September 10th | Begin testing components in project, start prototype |
| **6** | October 1st | Finish prototype; have at least one function fully running |
| **9** | October 22nd | Finish project; test all functions |
| **13** | November 18th | Prepare project for presentation, multiple tryouts |
| **14** | November 28th | Showcase final project presentation |

## 2.6 House of qualities

For this section, relationships of the customer needs and engineering requirements were made within a combined matrix called the House of qualities. The team evaluated the interactions with a positive or negative correlation to achieve a well thought out design.

The center grid containing symbols displays how strong a relationship each customer needs has in respect to engineering requirements. The roof (top part) displays potential connections and conflicts between engineering requirements. The importance number at the bottom was the overall importance after evaluating customer importance and engineering relationship.

This Figure 1 can be seen below.

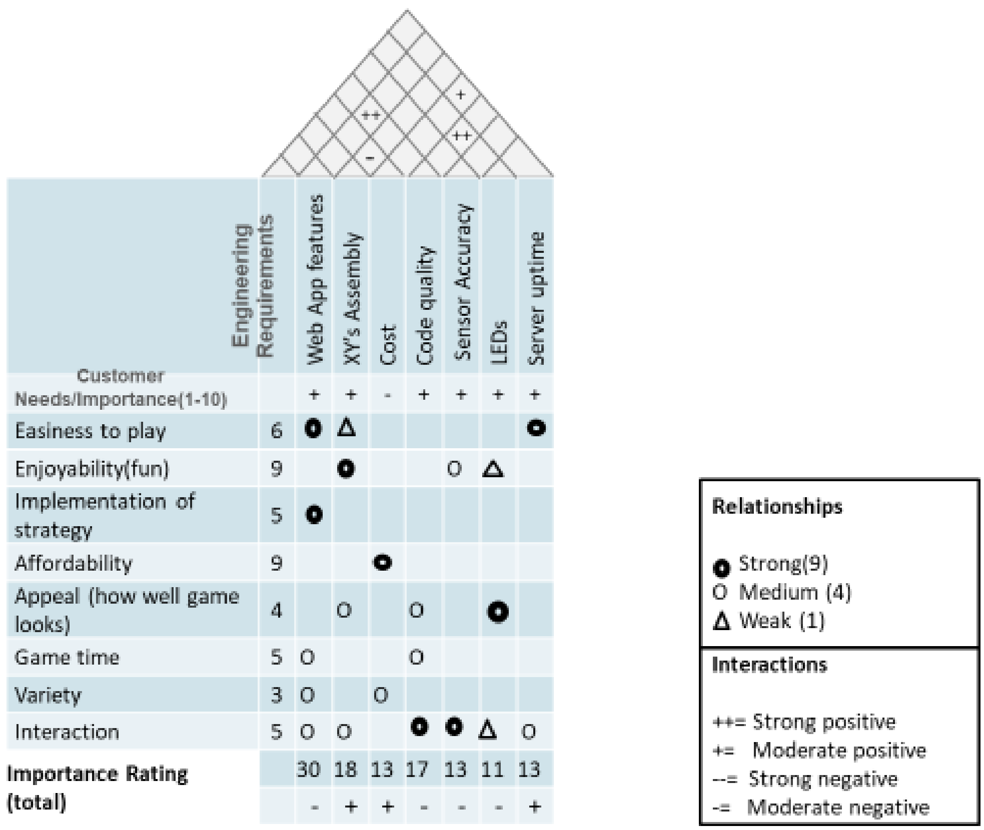


*Figure 1: House of Qualities Diagram*

## 2.7 Block Diagram

The Project Block diagram as shown in Figure 2 below, represents a high-level understanding of the various systems that was integrated within the project. It is a two-part application, in which a web app and a tabletop application was to interact with each other. Responsibilities were assigned to each person towards specific blocks of the project and the current status was kept track with weekly updates.

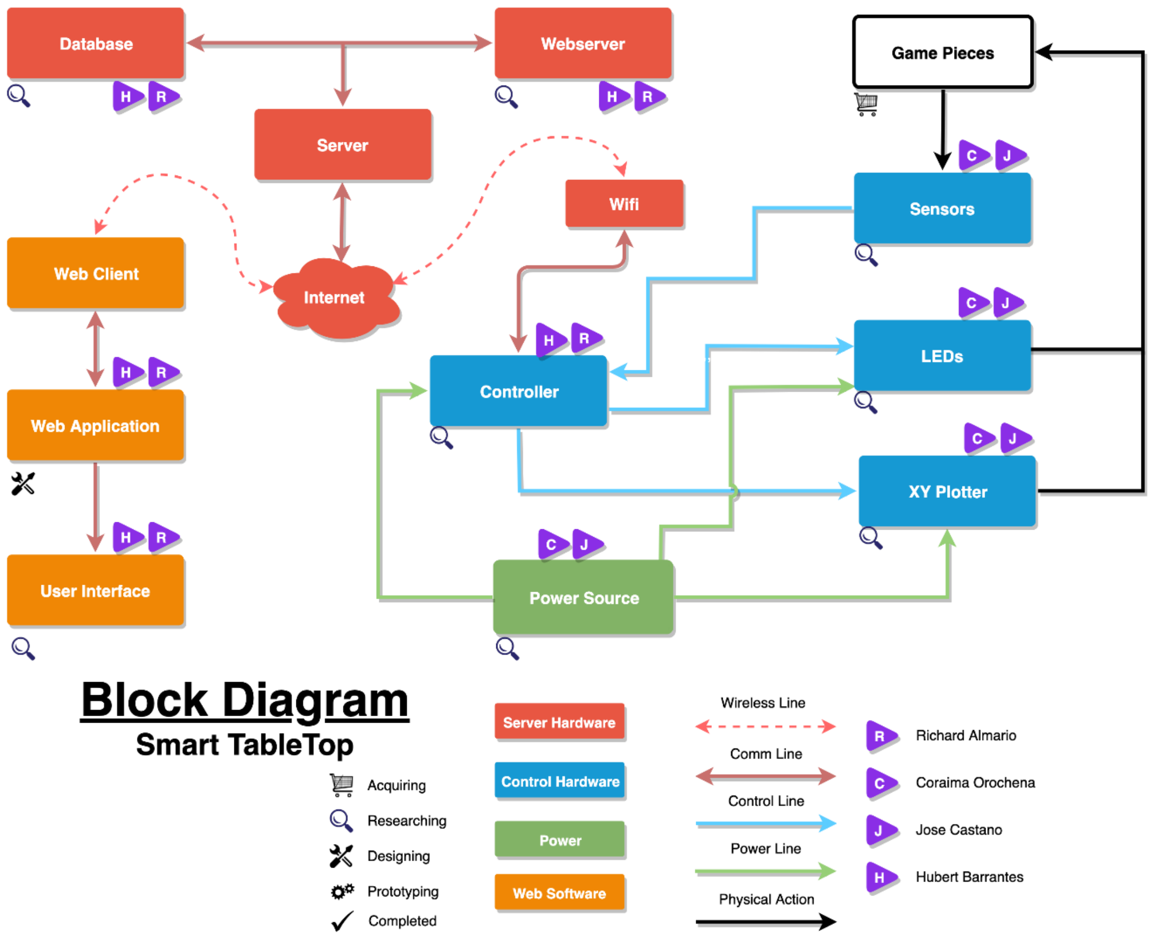


*Figure 2: Project Block Diagram*

## 2.8 Project Prototype

The project prototype was a 3D model representation of what the team believe the tabletop application was going to look like. This contains the different layers of the device that was going to be developed and integrated into the device shown in Figure 3 below.

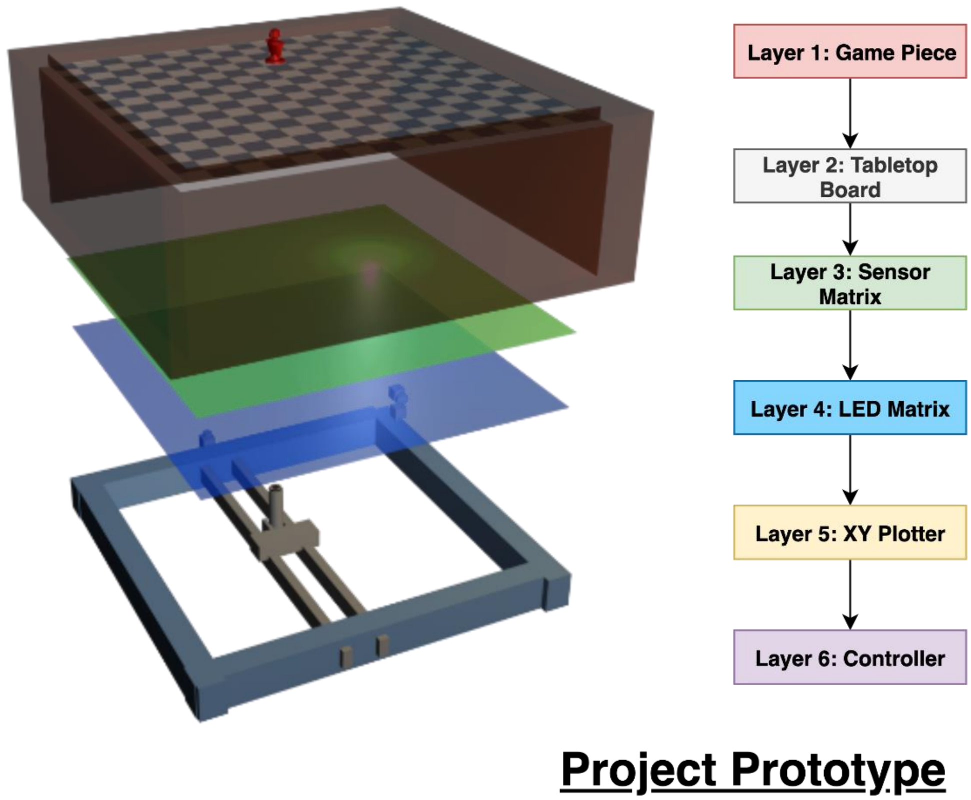


*Figure 3: Project Prototype I*

# 3.0 Research

This project could be a continuation from other projects that were made in the past. From there, information can be acquired from these documents and use them as guide for the making of the project.

## 3.1 Project References

For the group to develop a tabletop game that is dynamic and innovational, the group conducted research into existing projects for the purpose of gathering data and information that could be helpful. These projects have similar design or features that are good examples to us while the group developed a tabletop game. The group was going to look at the main ideas implemented into these projects that could be applied and improved them.

### 3.1.1 Magic Chess

Magic chess was a hands-free, voice activated chess board game that was built by another senior design students' group here in the United States. It has the option for the player to go against another player or the computer. One of the primary features of this game was the absence of a player's need to physically move a game piece as well as the inclusion of an intelligent chess engine that was allowed a player to play alone.

The structure of each game played has a simple series of event, when a new game started, a player accessed an easy-to-use interface via an LCD screen which presents the player a few simple options such as the number of players and the difficulty level of the computer. A human player then made a move by vocalizing a command, the board was to interpret this command via voice recognition software and used the electromagnet controller to carry out this move.

If a player attempted an illegal move, the board was informed the player via a message on the LCD screen and was going to wait for them to attempt another move.

This project has many similarities with their project hence why it was chosen for research and to improve on, although the project is not a chess game, it does used game pieces and controlled them using an X/Y board powered with sensors and moved the pieces without the player physically touching them via a web application built.

### 3.1.2 Telepresence Chess

Telepresence Chess board game was an interactive game that took on the classic chess game and presented it with modern flair. Chess has long been digitalized and been playable throughout the web but lacks the same gratification that a player got from an actual board and pieces in front of you which is what their project works on.

Each chess board recognized the game pieces in play and moved them without human interaction, the board recognized when a move has been made by a player and communicated through the internet to the other board what that move was. The board that received the move then automatically moved the piece for the player, just as if they are playing against another person on the same board.

This part was accomplished through a magnet implanted in each chess piece underneath the playing surface that was hidden to the user eyes, also it contained a Radio Frequency Identification(RFID) tag to uniquely identify each piece on the board which avoided moving a piece that was not called and to identify the position of any piece at any movement. Other features included saving games and loading previously started games which had piece locations saved before quitting.

This project related players the same real-life experience from an actual board and pieces in front of them without having to touch them. Their Radio Frequency Identification (RFID) was a very interesting aspect since it was excellent to identify pieces and their locations, this feature was to be considered in the project. However, the group has decided not to implemented RFID later on.

### 3.1.3 CyberChess

CyberChess was another project that has helped the group significantly with researching this project. It was made by another senior design group at UCF in Fall 2012. The biggest difference from their project and this project was that their project was another chess board game like the other related projects and that its voice activated like one of the other related games. The features int his game included being entirely voice activated to command the pieces with what to do and where to move.

The way they're able to do this was with a wired microphone. Like the other related projects, including this project it also utilized an XY plotter to move the game pieces around as commanded. This board also uses LED's, but not for same purpose as this project. LED's in the CyberChess project were used to just give notifications to the player as to what was going on with the game now. One other thing that the CyberChess project has that this project doesn't have was the use of audio to be able to communicate ideas to the player.

One idea that was really investigated in the CyberChess project and was going to be used in this project was the fact that they used reed switches to make their sensor matrix. Before researching into the CyberChess project the group was strongly considering using and only using hall-effect sensors to make their matrix and it was honestly going to turn out to be a big challenge for the group to wire each of the 256 sensors together and need additional circuitry to even be able to make it work. The CyberChess group discussed about how they were considering using hall-effect sensors for their project and how they preferred not to use them because of all the wiring and circuitry that it would require to make a matrix of them work.

That's when they mentioned reed switches and how they don't need a power source or wiring to work in a matrix. That fact was extremely helpful for the group because now they know that sensor matrix was going to be much easier to set up and using reed switches instead of hall-effect sensors which saved the group a lot of time. Assembling a reed switch matrix was going to be time consuming for the group since there was going to be 256 of them. So, if it took that long to be able to assemble a reed switch matrix. Imagine how long it would take for them to assemble a hall-effect sensor matrix.

Without having found this project beforehand it still would've been certain that the group would've found out about them, but there wouldn't have been much guidance as to how to build and set up the matrix. The CyberChess project gave the group a helping hand when it came to see an example as to how the matrix would be set up in a chess game in an 8 in by 8 in matrix. Since this project was not a chess game and was Dungeons and Dragons instead it was going to be a 16 in by 16 in matrix, but this project offered a great helping hand and blueprint as to how it would be designed for a 16 in by 16 in matrix also. The CyberChess project was a major contributor to this project.

### 3.1.4 Knight Light LED Chess

Knight Light LED Chess was another project like the other ones and has also helped contribute to this project. This project was different from the other related projects because it focused more on the aesthetic parts and for guidance of using the LED's to bright up the board.

This project was also made by a senior design group from UCF in Fall 2012, same semester as the CyberChess and as mentioned before even though they are similar the biggest difference between these two is their use for the LED's.

In CyberChess the LED's were used to just give notifications to the player, but not really for anything else and in Knight Light LED Chess the LED's were used to light up the whole board and to light up the path or spots as programmed for the game, which is the same set up and purpose that the group wants to set up the LED's for this project.

One of the other big differences between the Knight Light LED Chess project and the other related projects is how it also has a learning tool to teach players or help them out on how to make their move, especially for beginners in the game who are inexperienced.

The Knight Light LED Chess project just like this project are making these projects just for the goal of making something more fun and putting a different perspective on a games that were thought to be as boring or complicated to most people that were not familiar with the game.

The reason the Knight LED Chess project was helpful for this project was because of how the LED's were being implemented for their project. It is an 8 in by 8 in chess board and the LED's were being used to light up pathways as to where the player should move their game piece. As the Knight Light LED Chess project provided guidance to new players or beginners to the game.

For this project the LED's were desired just for the aesthetic purposes, so it looked nice and gave more life to the game. It was used to light up the path to where the player chooses to go and was to light up on a character when it's their turn to make a move on the game. Chess was a strategy game, while Dungeons and Dragons was also considered a strategy game, but was more for fun and games, than being complicated like chess.

Another thing about the Knight LED Chess project that was helpful to this project was the explanation that was given to how to set it up. It gave many helpful details and was thoroughly explained in every step how it's done to LED's themselves to the LED drivers. It was also helpful to know what LED drivers were used and how they were implemented to power the LED's. The Knight Light LED Chess project turned out to be very useful to help the group with this project. Just like the other related projects it uses an 8 in by 8 in board and it was a chess game, but it gave lots of guidance and help as to how this project can be done.

## 3.2 Relevant Technology

This section outlines all research done regarding technology that was implemented to the design to make it function as described by the specifications and executive summary. It's important to look at all the different technologies that can be useful for the purpose of the project to know the options and what can be used in case it does not work with the main preference.

### 3.2.1 X/Y Stage

The X/Y stage was a very important part of the project; therefore, the group has evaluated spending a good portion of the budget on it to make sure it was made as well as possible. Two possible options for the X/Y stage were either purchasing a pre-assembled stage or for the team to assembly it.

Assembling the X/Y stage could be very time consuming and complicated since it requires some mechanical engineering knowledge. It also needs to meet the requirements and specifications that were laid out in the requirements sections and most important needed to remain in the budget set for the project. As previously mentioned, the group was likely not assemble the X/Y stage due to potential complications resulting from it.

A pre-assembled X/Y stage was the most ideal option for the project since there was no mechanical engineering students in the group. The pre-assembled stage met specifications, however, budget limit was the concern.

### 3.2.2 Wi-Fi Communication

Wi-Fi is a technology that uses radio waves to provide network connectivity. A WIFI connection is established using a wireless adapter to create hotspots. A project was using this capability to exchange information between the physical gameboard and the digital board.

Wi-fi worked by allowing a device to send and receive information via radio waves. Every device has an adapter that translates data to radio frequency signal. This signal was then transmitted via an antenna to a wireless router. Once the router received the signal, it decoded it and sent/received data from the internet in a local area network, which is called LAN.

Once all steps were performed, the router sends data from the internet/other devices back into the original device. For the project, the group used a wireless LAN to transfer data from the microcontroller to the server in order to execute commands as well as synchronize game states between two users. The group was considering using this operation for a multiplayer mode gameplay if applicable. For any other mode, where there was only one player exploring or just playing the game, no information synchronization was required since the board was using sensors that was letting the microcontroller the current state of the game.

The Wifi Alliance is a global non-profit organization that was formed in 1999. They deal with products from different manufacturers that are certified by the basis of IEEE 802.11 standard for the operation of many wireless devices. Their goal was to organize, stablish and enforce standards for Wi-fi enabled devices and products.

Security standards were maintained according to the WPA and WPA2 security standards with EAP authentication standard. The Wi-fi certification on the device can be obtained when these standards are maintained by member organizations of the Wi-FI alliance. This allows for the product to use the Wi-fi seal as a trusted device for backward combability. Nonetheless, if a product uses the Wi-Fi seal, it can then function with other devices with the same certification as well as with other devices following recently outdated standards. The project was designed with these parts to avoid any difficulties when integrating parts to the system. By comparing when Wi-Fi was established back in 1999 until now, prices have lowered and keep on lowering which allows Wi-FI communication to be used in more systems. LAN(Local area networks) allowed for devices to be connected wirelessly in many environments.

There were other wireless connections the group considered using such as Bluetooth but Wi-Fi had many advantages over it such as increased transmission range and power, more speed and can manage more devices. One downside of implementing Wi-Fi into the project is that Wi-FI signals can be interrupted by neighboring access points which causes a loss in connection or results in slower internet speed. These interruptions are caused because of overlapping channels in the 802.11g/b spectrum. As a result, data transmission is slower as inaccuracies appear because of the decreased signal-noise ratio.

The tabletop game does not require high performance equipment, it was not going to transfer long distance data and not use it with near sources of interference, therefore there was no concern with using Wi-FI technology. One concern for using Wi-Fi was going to be power consumption and configuration. If the system has a large power source, it was going to be able to maintain communications with devices without any problems because Wi-Fi has strong security protocols which protects users from attackers, although security threats were not a concern to the group since the project was a concept-proof type and not commercial so this was not going to be considered more of an extra rather than a must have feature. The biggest obstacle the group faced when setting up a wi-fi connection was the Wi-Fi communication between the microcontroller and the devices that were around. Team members had to do more extensive research on the communication, one option they found was using automated configuration features, this allowed for a simpler integration of parts.

### 3.2.3 Bluetooth Communication

Bluetooth is a wireless technology that has been around for more than 20 years. It allows many devices to interact, sync and connect without the need of setting up complex passwords and networks. These days, Bluetooth is everywhere from mobile phones to laptops as well as car stereos. It supports a wide many device and it can be set up in only a few minutes.

More recent Bluetooth versions have made it possible for users to use hands-free phone calls through a mobile device and other examples like connecting to music playlists. Bluetooth technology can simplify tasks, for example, with a Bluetooth enabled printer, one can connect wirelessly with a desktop, laptop or mobile device to print documents. It is also possible to sync a table-style device with a wireless keyboard like an Apple iPad or kindle fire.

Potentially using Bluetooth has advantages as well as disadvantages, the main advantage was connection establishment to be very easy, this can be crucial since having an easy connection can give the group more time for other parts of the project such as the interface between the microcontroller and server which could potentially take some time. It also has a lower power consumption unlike a Wi-fi connection. Other advantages include a better range than infrared communication, used for voice and data transfer and the technology can be adapted by many products like printers, webcam, GPS systems, keyboard, etc.

Just like it has advantages over other wireless technologies like WI-fi, it also has some disadvantages. The main one found was that it is very slow compared to Wi-FI,, its data transfer rate will be in the kbs to few mbps compared to a new Wi-Fi module which uses 5GHz spectrum with MIMO technology to achieve speeds as high as 1.3 gbps. This can get complicated when transferring large files so it could potentially make the project interface slow.

Other disadvantages are that it has a very short range, a signal can be lost if the displacement is roughly 10-20ft and it was not ideal to use this type of connection worldwide.

Regarding the project connection, the group has decided that Bluetooth was going to be used as a secondary, backup function. The group was going to use Bluetooth when bug testing as well as for a prototype to test connections. If the team members cannot stablish a Wi-Fi connection, then Bluetooth was the technology to be used. The main problem was that Bluetooth cannot be used to play the game because most of the functionality of the device runs through the database and the need for a connection to the internet.

### 3.2.4 Motors

This section shows research done on many types of motors that could potentially be used to build the XY plotter.

Stepper motors are DC motors that move in discrete steps. They have many coils which are organized in groups that are called “phases”. When each phase is energized in sequence, the motor will start rotating one step at a time.

They come in three forms which are permanent magnet, variable reluctant and hybrid but their functionality is the same. Each winding set is energized one after another, when they are energized it makes the rotor align its own magnetic field with the one that is being produced by the windings.

These types of motors were relatively easy to obtain and less expensive than others. it also has the advantage of having simple controls and it eradicates a closed-loop control system to determine location. Because of these features and the easiness to get the group has determined using stepper motors will be the first option for building the XY plotter

A direct current or DC motor converts electrical energy to mechanical energy. It consists of a rotor, an stator, an armature (coil or wire) and a commutator with brushes.

It is equipped with magnets, either electromagnetic windings or permanent magnets that produce a magnetic field. When current flows through the armature placed between the south and north poles of the magnet, the generated field by the armature communicates with the field from the magnet and applies torque to it. In a DC motor, the magnet forms the stator, then the armature is placed on the rotor and the commutator switches the flow of the current from one coil to another. Once this is finished, then the commutator connects the stationary power source to the armature via the use of conductive rods or brushes. Furthermore, DC motors operate at a fixed voltage.

Acquiring DC motors and spare parts is very easy, they also include easy speed regulations. After doing some research, the group has determined that DC Motors was going to be a second option in case the group no longer implement stepper motors.

Servo motors are rotary actuators that allows a very precise control in terms of angular position, velocity and acceleration, these capabilities other regular motors do not have. They are different from DC motors and stepper motors because they have a controller as well a set of reduction gears in their housing. It makes use of a regular motor and pairs it with a sensor for position feedback. In these types of motors, the controller is the most sophisticated part of it, since it was specifically designed for their purpose.

There are two different types of servos: Continuous and standard. Standard servos go to a position and keep it on hold until the user sends a new location, whereas continuous servos go to a velocity and hold that velocity. All servo motors require a closed loop control system which consists of external sensors to give it a new position or velocity to match it.

Using servo motors has some advantages for the project, it has minimal noise production which the group has found important since the team want the users to have a focus, in game experience and if there is a lot of noise this could distract players. It also has high seeds, works very well for velocity control.

### 3.2.5 RF-ID

Another type of technology relevant to use and was considered being used in this project is RF-ID. RF-ID stands for radio frequency identification. It's another form of wireless communication like Wi-fi and Bluetooth which was previously discussed. RF-ID uses the radio frequency portion in the electromagnetic spectrum to work, with the use of electromagnetic or electrostatic coupling it can identify objects, like animals and people or other objects. That is how it would be used in this project to be able to identify the different game pieces and communicate to tell them what to do in the software.

An RF-ID system includes three components a scanning antenna, a transceiver, and a transponder. The transponder is the RF-ID tag which includes a microchip, memory, and antenna. An RF-ID reader was a choice to consider for this project because it was portable or can just be attached to the board. The way the reader was going to work was by transmitting signals using radio frequency to be able to activate the tag. The tag when activated will then be able to wave at the antenna which then it will be translated into data.

For this project, any of the two types of RF-ID tags were used. The two types were active and passive. It was preferable for the project to use a passive RF-ID tag because an active RF-ID tag required that it has its own battery or power supply. The passive RF-ID tag does not require any batteries or power supply and instead relies on getting power from the antenna, and the antenna supplies power to it by making a current that goes into it through an electromagnetic wave.

Another type of RF-ID tag was the semi-passive RF-ID tags and this one was a mix of both active and passive filters. The semi-passive RF-ID tag uses a battery or some power supply to power the circuitry and then the communication is processed through the RF-ID reader. A semi-passive RF-ID tag would be useful but is not necessary for the purpose of the project and is probably more expensive than either an active RF-ID or passive RF-ID. Those are the options for the project, but they are expensive choices.

There were also three types of RF-ID systems, they are low frequency, high frequency, and ultra-high frequency. There was also microwave frequency, but it was not necessary for the purpose of the project. For the purpose of this project, only low frequency was applicable and high frequency can be considered but is not necessary. Low frequency just goes into short ranges from just a few inches to less than six feet at most, which was perfect for the purpose of the project.

High frequency ranges from just a few inches to several feet, which was still useful for the project, but not necessary and probably more expensive than low frequency. Ultra-high frequency and microwave frequency were used to read things that 25 or more feet away. That is extreme and not necessary for the purpose of the project because everything that would need to be identified and read in the board is just inches away from each other.

RF-ID was something definitely being considered to be used in the project, but the main reason that the group was not using it as the main source of communication was because of how expensive it would turn out to be and how complicated it may be to set up compared to the other options of Wi-fi and Bluetooth. It was a very good idea and a good technology that would be useful to identify the game pieces in the game. If it was too expensive; too hard to set up, then there was no need to get it or use it when there were cheaper options and easier ways to set it up.

# 4.0 Software Design

The overall software design was sectioned between the Web application, the database, and the Microcontroller application. The Web application shall be always synchronized with the current state of the database, and two controls within this system was the end applications of the Web app and controller app. Each can influence the database, and since there can be multiple users performing different actions at various times, a need for a non-blocking application was key to minimize possible errors between the applications.

## 4.1 Initial design

The initial design was the starting idea for how the software was to be structured. With this project, the application must be responsive and be able to receive multiple actions with a minimal error margin. These events can happen at any time, while the only action that was set to be synchronous is the movement of a game piece.

When it comes to movement for most tabletop games, it expected a defined two-part action of lifting a game piece and placing it down in a different location, so any movement action within both applications was expecting a complete movement action. Other actions such as pinging the map, doing a spell effect, and adding a new game piece can be done asynchronously.

### 4.1.1 Web App Design Flow

The design flow starts from the very beginning with the user loading the URL of the webpage. The intended design was to go for a single page dynamic website, that was not require the website to redirect or reload the page for changes to update for the client. Since multiple users can connect to the web app, there was a need for a front-end login system, with three types of users. There was the player, the Dungeon Master (DM), and the admin logins. Once past the login page, the 1st design in mind was to have a single room where all game actions can take place. Future iterations of the application could have a multi-room lobby system, but this design was for a single tabletop in mind.

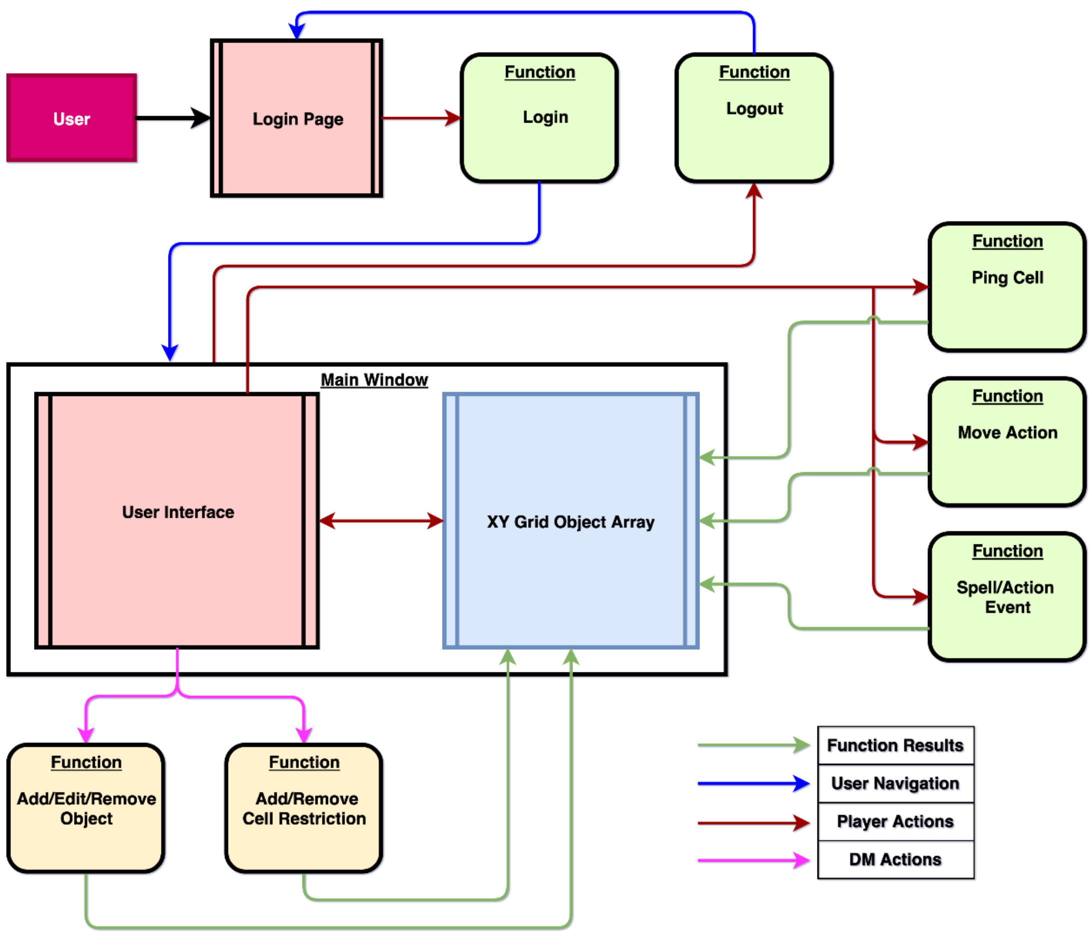


*Figure 4: User Software Block Diagram*

Once inside the web room, the user interface was clearly portrayed different functions that the player can perform. In one section there was a 10 in by 13 in grid, showcasing the main playing area. Another section contained the player's character information, and below that section has button functionalities to affect the game board by moving their character, pinging the map, or performing character abilities that influence the game board. In the top right will be a logout functionality. This overall design is portrayed in **Error! Reference source not found.** on the previous page.

A DM was having extra functionality that a regular player can't perform. Their privileges allow all the same functionality of a regular players, with the added benefit of being able to move any character's or tokens on the game board. They also have an assortment of extra functionalities they can perform. They can designate grid spaces to become impassable or clear at any time, to simulate a map change. Some examples include exploring a cave system and finding a dead end, simulating a trap, or supernatural obstacles. The end effect is to have a grid space become blocked, so that the player will have to be creative or navigate the obstacle.

Another features the DM was able to perform was to add or remove tokens on the game board. If a token was added on the tabletop application, a prompt will display on the DM's user interface to confirm a new token. If accepted, this token was given attributes which contain the name, health points, armor class, and speed. This feature implementation was further explained in detail in section 4.3.5, while how this data is handled is within section 4.5.4.

### 4.1.2 Tabletop Application Design Flow

The design flow for the Tabletop application depended on what microcontroller the group was utilized. The first choice was a Raspberry Pi, but after careful consideration on pricing and the functionalities the group would need, the group has opted for the Mega Arduino 2560.

The Tabletop application was a two-part software scheme. One was communication API, where event communication and data manipulation were performed, and Input and output block, where sensor and control actions was performed. The application was in sync with the database and socket connections were used for functionality transmissions.

When a move event was transmitted through socket communication, it referenced the database, retrieved the specified path plan, and sent this in a proper format to the controller to perform the XY plotter positioning moves. When a ping or effect event was received, this translates to a LED action where the computer referenced the database for the proper LED array, and sent the data to the microcontroller's LED control for proper utilization.

This representation of the various blocks of the software is displayed in Figure 5 below.

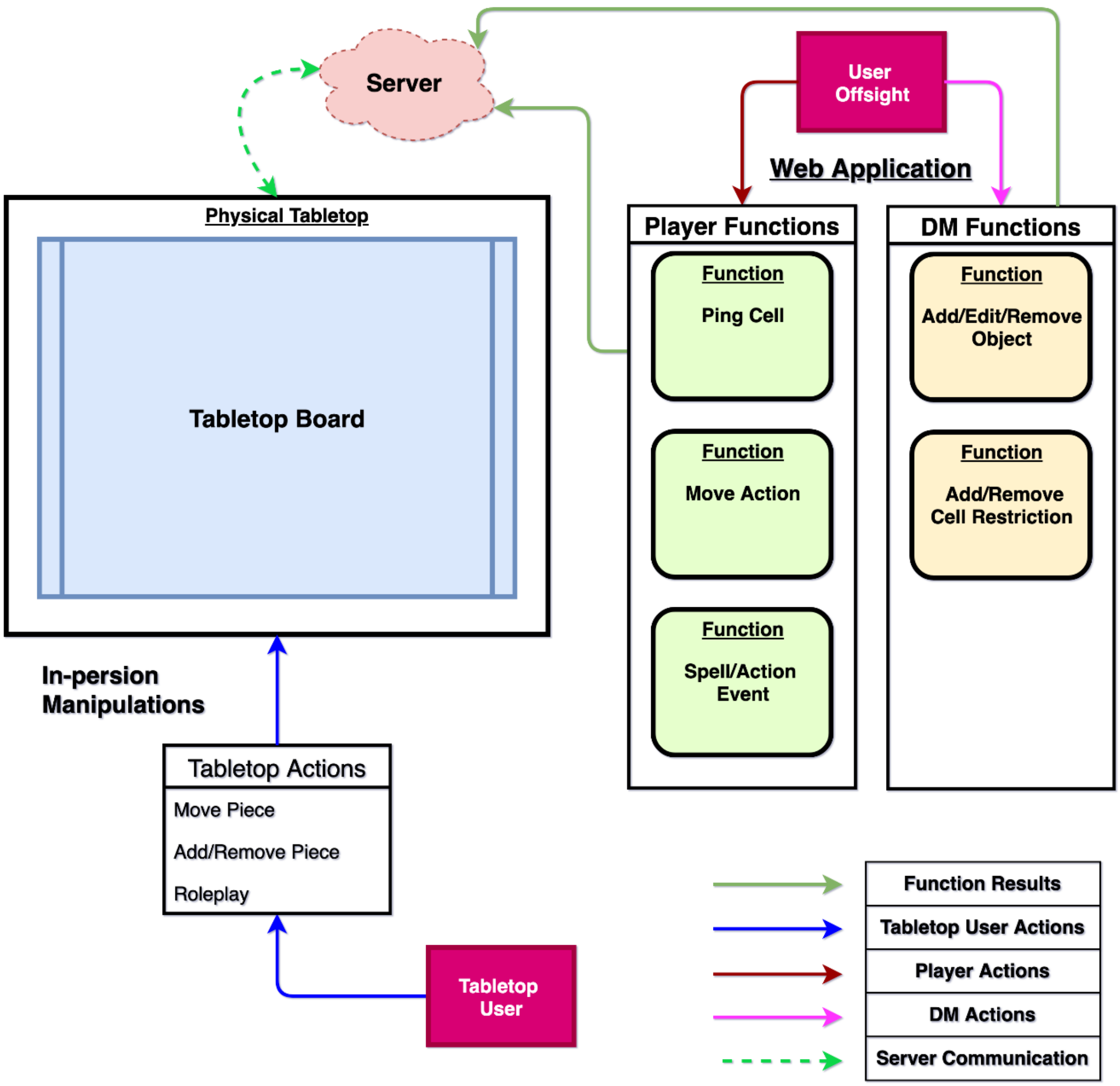


*Figure 5: Tabletop Block Diagram*

### 4.1.3 Front/Back End Design Web Frameworks

Various frameworks were researched for web application, and the tabletop application. The group wanted a responsive and well documented framework, and research went into Model, View, and Controller (MVC) frameworks. Another line of research was done into Model, View-Controller, View-Model (MVVM) Frameworks as well.

They both serve as very reliable web frameworks for the job that the group wanted done, but when it comes to specific functionalities, the group wants to go with the best framework in the beginning to reduce development time and improve scalability. The primary goal was to create a single page application and have real-time communication between the web app and the tabletop app.

The initial choice was to go with an older model such as MVC, where the framework was designed for modularity and ease of development. One example of an MVC framework was CodeIgniter. It was widely used framework that can do the job for the web app, but primarily runs as a WAMP stack. For the project, the group has decided for higher performance and asynchronous communication.

This direction pointed me towards using a backend that utilizes node.js. Using NodeJS allowed to do both the front and backend with only JavaScript and without the overhead of other programmers having to learn backend-based languages such as php. The database was involved using MongoDB over MYSQL, due to JavaScript being a good cross platform language and not having to learn SQL was another perk of using the database choice.

Ultimately, when building the web application. The software that was used was a variation of the MEAN stack. It implemented NodeJS as our webserver which is used when building a MEAN stack but instead of using MongoDB as the database the group opted towards using a local database storage by using JSON files to store all data. This changed was made because a local database is free, more secure and does not require internet connection to access it. Instead of using AngularJS as the front end, the group decided to implement Angular since it is a more updated version of AngularJS and has many more improvements compared to AngularJS

## 4.2 User Interface

The web application user interface was one of the most important elements on the project since it was the portal in which all users will interact with the board and receive feedback.

It displayed all the functionality in such a way that it can be easily understood and performed with little to no training, as well as being a hub to receive all changes that occurs on the main tabletop application. There were different ways to interact with the board system, for the project the group was considering two different approaches and see the advantages/disadvantages of each when it comes to how users were able to interact with the system.

The first approach was command line interface, out of the four options discussed this was the oldest method of use interface. It involved the computer responding to commands that were typed by the operator.

For the project, this type of interface has the drawback that it required the operator to remember a range of different commands, this makes it very difficult for novice users to manage. Therefore, unless the other user interface methods do not work, the group was not going to implement this type of interface.

The second type of user Interface was Graphical UI. It was also referred as WIMP since it used Windows, Icons, Menus and Pointers. The operator used a pointing device (such as a touchpad, mouse) to control a pointer that was on the screen which then interacted with other elements on screen. This allowed the user to interact with devices with graphical icons and visual indicators. Using this type of interface has some advantages as well as disadvantages which are explained in **Error! Reference source not found.** below.

**Table 23: Graphical UI Advantages vs Disadvantages**

| Advantages | Disadvantages |
|---|---|
| – Relatively easy to use, ideal for a beginner. | – GUIs take a large amount of hard disk space compared to other interfaces |
| – Easy to explore and find way around system using the WIMP/GUI interface method | – Need significantly more memory(RAM) to run compared to other type interfaces |
| – No need to learn complicated commands | – Use more processing power compared to other type of interfaces |
| – Help system available within the WIMP interfaces | – Slow for experienced programmers.<br>– If experienced, CLI interfaces are better |
| – Allows you to exchange data between different software applications | |

## 4.2.1 Web Application User Interface

A client/server program which the client can run in a web browser is called a web application. The web application was delivered to the client through the internet using a browser interface. This design was delivered to the client through internet using a browser interface.

Since the web application was running within web browsers, there was no need to develop web applications that were compatible with multiple platforms since the application depended on the browser and not the OS. There was a long list of frameworks and libraries the group can use in order to develop the web application in an efficient, fast and convenient matter. After doing some research, the group was discussing the best UI libraries that were found that could potentially use for the web application.

The first one found was Kendo UI, it is a JQuery-based development framework. It was a collection of scripts, styles and images. When installed, the team members received a lot of JavaScript files, stylesheets, images into the project and Kendo UI leveraged them. The group was considering using this UI because it could build modern web applications very quickly and very easily as well. It has over 70 UI components, a customizable and preconfigured theme and built-in support for Angular 1.x. Kendo UI provides all the tools needed in one package so there won’t be any need to download many libraries to make the web application look better. Since Bootstrap was a good candidate for a framework for usage. After doing some extensive research and found the advantages and disadvantages of using it which will be discussed below in Table 24.

**Table 24: Bootstrap Pro's and Con's**

| Advantages | Disadvantages |
|---|---|
| Great Standardized platform with all components and basic styles needed | Styles are verbose, can lead to lots of HTML outputs not perfectly semantic |
| Supports all major browsers | JavaScript is tied to JQuery |
| Fixes CSS compatibility issues | Requires lots of overriding styles or rewriting their files if many customizations were done. |
| Consistent UI, looks good out of box. | Websites look the same if colors and styles have not been customized too much |
| Lightweight and customizable files | |
| Designed with responsive structres and styles for mobile devices | |
| Several JavaScrip plugins included | |

| | |
|---|---|
| Good documentation and community support | |

The second framework the group has found was angularJS, this HTML was very good for declaring static documents. It allowed the group to use HTML as the template language and extend HTML's syntax to express the application's component clearly. AngularJS was great because its simplified application development by presenting a higher level of abstraction to the developer. On top of this, it has an improved server performance since it supported caching processes. This framework although it has excellent tools to help the group with the project, it also has disadvantages that come along. The first one being that it was difficult to learn, it has a limited documentation available which affected the learning process. It also has not been built for mobile devices, since the project was featuring a web application, this framework could potentially not work.

The third and last framework the group found was Bootstrap, it was an open source tooklit for developing with HTML, CSS and JavaScript. It allowed the group to quickly prototype ideas as well as build the entire web application with their Sass variables and mixins, extensive prebuilt components, responsive grid system and powerful built on JQuery.

Lastly one more framework the group found helpful was Vue.JS. It was an open-source JavaScript framework great for building user interfaces, websites as well as single page applications. It featured an incrementally adoptable architecture that focuses on component composition and declarative rendering.

They found many good reasons to consider using Vue.JS as one of the frameworks for the web app. It provided very good performance, very easy to start with and offers plenty of essential features out of the box thanks to its component library based on HTML, CSS and JavaScript. It has a routing and data management covered by official libraries. It was very flexible to design the app structure any way the group want it and thanks to its reactivity, data binding between HTML and JS code was easier than ever.

Just like it has advantages, the group have found some disadvantages for using Vue.JS. The first one being that it was a relative new platform and therefore it has a small community. It was not as popular, and most of its users were non-English speakers, this could be troublesome in case the group need support. It can also have Over-Flexibility since it has so many options within its framework but for the project since it was considered a bigger project it can over-complicate it and lead to more errors and irregularities within the code, which can make the project delayed and more expensive

Ultimately, the group found and utilized Angular materials for our user interface when showcasing the project.

## 4.2.2 Layout Design

The Layout of the software had the login screen, which on successful user verification, lead to the main user interface. The initial design of the interface had three main sections. These three sections have subsections for proper layout of the different functionalities and had to expand or contract with all the content being displayed properly. The range of display ratios accommodated for different devices, ranging from desktop computers to smart phone displays.

The biggest section layout was 10 in by 13 in XY grid display. This display was contained all the game objects that are currently on the board, such as creature's, user tokens, and grid space obstacles. Each grid square was interactable, depending on the selected action or pointer event. When a game object was selected, a pop-up module window appeared next to that object if it can be modified by that current user. Selecting the object also displayed any accessible data on the user information panel, which was explained further ahead. Each game object was color-defined by their classification, in which green was the current user's game object(s). Blue defines game objects controlled by other players, while red and yellow defines hostile and friendly NPC's respectfully.

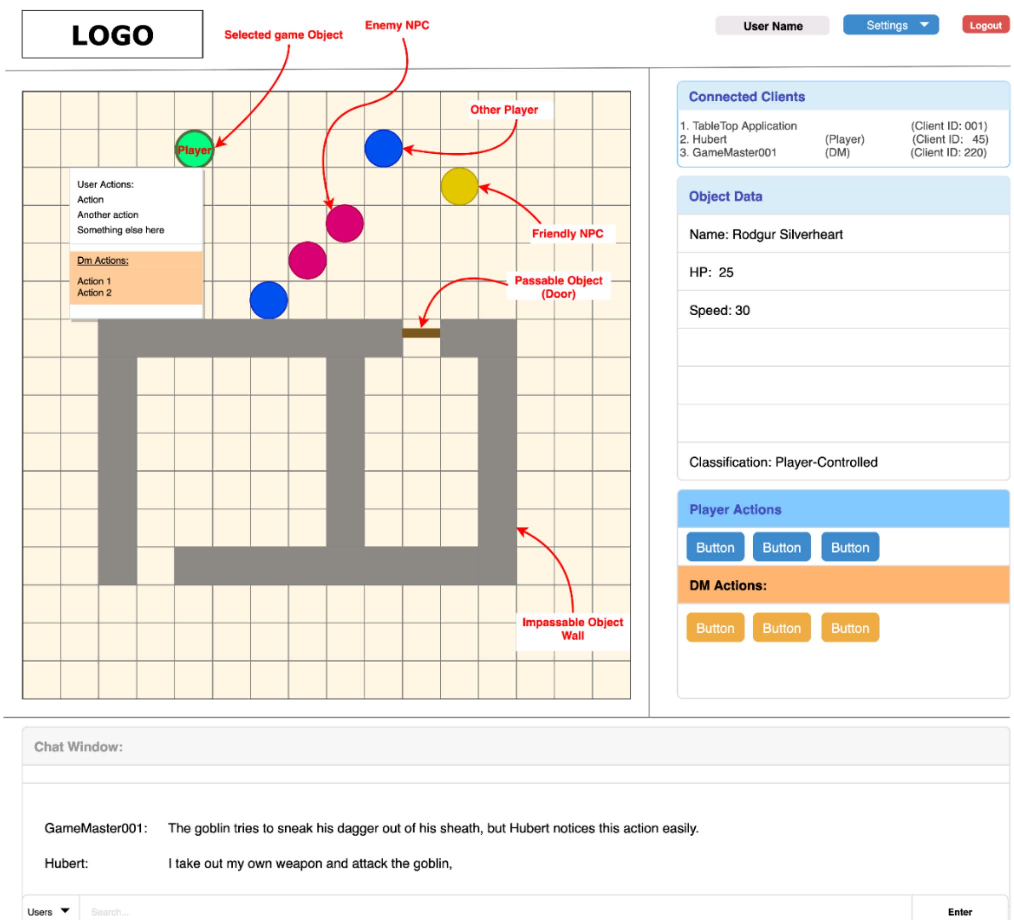


*Figure 6: User Interface (Version 0.1) Diagram*

These actions events were within their own section, which was called the game control panel. This panel has separate sections for the types of users. Regular players were only using their specified actions, such as shown in **Error! Reference source not found.**, while GM's had an extra set of tools that they can use. These extra tools were not visible to the regular user.

Another main section was the client connection window. This section kept track of all connected clients and updates its list when a new user logs in or when a user logs out. This had to be done by either closing the window or pressing the logout button that was placed at the top right of the browser window. This window served another purpose to show what type of user privileges that user currently holds. User permissions are fully covered in section 4.3.5.

The last main section was the user information panel, which displayed to the player the character data of selected game object that were on the board. The regular user can only see information on objects they own, while GM's have access to all available object data that is on the board at that time. The representation of this user interface is displayed in **Error! Reference source not found.** on the previous page.

This diagram was the first iteration on how we would build the main interface where most of the events was taking place. The theme and various elements were not considered final, and some functionalities was removed or added as development progresses. Implementation of front-end libraries, such as Bootstrap, was helping speed up this task and helped in refining the user interface to its final version.

When it came to user's arriving at home page of the web application, they first arrived at the login screen, where proper credentials were needed in order to progress any further. This window contained the basic requirements of a login screen, which included the username, password, a confirmation button, and a register new user button. When a new user was made, all passwords were hashed/salted to secure the user's account and the user was able to log in right after registration.

### 4.2.3 XY Grid Array

There are 130 cells in the grid matrix, each has several properties including mainly X/Y coordinates. An array was designed to holding the statues of all 130 game pieces squares. When designing this array, it was set up so that a "1" meant that a game piece was in that location and a "0" meant the location was empty. By applying this reference, the microprocessor was very if a game piece was in the destination place or not. If the destination place was empty, then the game piece was moved directly to it and if the destination place was full the game piece that was occupying it needed to be removed first.

This method of using a "1" or "0" for every statue in each square was very common and straightforward which is why it was being considered. The group may however implement a more sophisticated approach that could allow more flexibility in the game engine.

The amount of game pieces or objects on the board could possibly reach the amount of grid space we have but it was not a typical number a game usually goes through. It was most likely reach an estimate 20 different game objects in one session. Below is **Error! Reference source not found.**, with a graph displaying the 1 and 0 method for empty and full squares. For each empty space a 0 is in place.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | 1 | | | | | 1 | | | | | |
| | | | | | | | | | | | | | | | |
| | | 1 | | | | | 1 | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | 1 | | | | |
| | | | | | 1 | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | 1 | | | |
| 1 | | | | | | 1 | | | | | | | | | |
| | | | | 1 | | | | | | 1 | | | | | |
| | | | | | | | | | | | | | | 1 | |
| 1 | | 1 | | | | | 1 | | | | 1 | | | | 1 |
| | | | | | | | | | | | | | | | |

*Figure 7: 10 in by 13 in Grid Array*

If there was a game piece in the destination square, the database needed to remove it. This function will be called "MOVE" by the main program. It received three different values necessary to remove the game piece that was in place to the graveyard. The initial and final coordinates for the game piece were discarded as well as the initial coordinates of the XY plotter. It started off by first receiving the current location of the X/Y plotter. Secondly it received the coordinate for the piece in hand, with this coordinate now the software can calculate how far it must go in the X and Y direction and the order in which it was going to make those moves. Once it followed these steps the X/Y plotter was positioned directly under the game piece to be discarded. Once this occurs the magnet was remained in an up position and remained there until the game piece has been discarded.

For the graveyard, their coordinates followed the same steps as when a game piece was removed by calling a MOVE function destination. The X/Y plotter then moved the game piece to the graveyard. Once this process was finished, the memory that contained the X/Y coordinates location was updated to the graveyard's coordinates. After this the software can now move into the empty statues.

Although the move function was implemented for moving pieces to destinations that were already in place, for the game itself game pieces will be moved but not into each other's destination since the game works in a way that it was a cooperation-type game rather than an elimination-type. For the empty statues, they were reached when either the destination statues was empty and the program moved it there directly or the destination was full and the program executed the previous program described to move the game piece to the graveyard and make it empty. The MOVE function will retrieve the initial coordinate of the X/Y plotter as well as the beginning and destination point of the game piece to move it to an empty square. This function was very useful into the game since game pieces were moving to new empty place throughout the game.

## 4.3 Game Piece/Action Control

Game Piece and Token control was a cornerstone requirement between the web application and the tabletop application. All actions and events were associated with an object.

### 4.3.1 Piece State

A game piece or token was in various modes to represent its current state. These states are displayed in Table 25.

**Table 25: Game Object State Table**

| State | Description |
| --- | --- |
| **Standby** | The object is waiting for the next event. |
| **Executing** | The object is performing an event. |
| **Move** | The object is in motion. |

When an event function was executed for a game object, the database was referenced for what state the object was currently in. If the standby state was confirmed, then the type of event was performed as well as updating the object to the execute state. If this event was a movement action, the game token and the tabletop piece perform verification if they were in their proper locations on the board, then updated their object to the move state. After the performed event, a finishing action was prompted the Tabletop app to update the game piece state to the standby state. Further detail on placement verification and movement was explained in their dedicated sections.

### 4.3.2 Piece Placement

Tracking piece placement and new game pieces were considered single actions. A new detected piece on the tabletop grid square was prompted a notification window asking for confirmation that this was a new game piece. Implementing movement of a game piece was done via dragging and dropping the token which was explained more thoroughly in section 4.3.4.

Piece placement was verified in through the tabletop application, but all placement data was referenced through the database. The reason why the tabletop app was the main device was that the board was where most of the action took place and where game pieces were interacted with without having to verify the server beforehand.

### 4.3.3 Piece Selection

For this project, one of the requirements was to make sure real pieces moved without the player touching them. The pieces were placed through the board and the player was able to move them by using the web application. The web application needed to be synchronized with the tabletop application. For any type of changes the player made the database utilized, where it recorded where the piece was placed and its movement and repeat. Once the game was over the database cleared all the game actions.

**Table 26: Game Piece Verification Comparison Table**

| | Magnetic Assumption | RFID |
|---|---|---|
| ***Pros*** | • Magnetic fields can be utilized in movement and sensing<br>• Easier implementation | • Unique identification<br>• Identification between data and object is assured. |
| ***Cons*** | • Piece data can possibly reference the wrong object | • Extra API utilization<br>• Can only be used to sense the game objects<br>• Extra expenses towards chips and chip readers |

The player was able to select their game piece from many options given, once the piece was selected the database will record this information and as previously mentioned, record all movements made by the player. There were a few ways to verify that the selected game piece was synchronized, and this was represented in **Error! Reference source not found.** shown on the previous page.

The first solution was magnetic assumption, where the game pieces locations were saved in into the database, with the game object data associated with that grid position. Whenever there was a change to the objects data, the database referenced the location to verify if that selected object for change was in its correct location on both boards. This was done by making checking if there was magnetic field triggering the sensor on the grid position, as well as checking the array position on the web app. This verification process made sure that the state of the board was consistent and reliable between the devices.

The 2nd solution was the RFID, where each game piece had a RFID chip at the bottom of their base. This can absolutely make sure that the object data was associated with the correct game token or piece. This direction though introduced another API that would have to be programmed to read the RFID chips and added a layer of difficulty to identify various game pieces. This also meant that that the group would need chips for each game piece and the readers on every single grid square, which increased the expenses dramatically.

The chosen solution involves using magnetic fields to detect the various pieces that were placed on the physical board. This meant that game pieces that have been placed on the board assumed to be in their correct positions to match the game data that is associated with them. Making sure that the game pieces were matched correctly is solely up to the group who are currently playing in that game session with the physical tabletop board. The algorithm made sure that game pieces were in their current positions cannot differentiate between different magnets that were placed on the grid spaces. Even with this drawback, using magnets allowed the game pieces to be manipulated with the XY plotter that were placed beneath the board.

The magnet chosen for our project and used in our showcase was a 250 newton electro magnet which was strong enough to move the piece on top with the help of the XY plotter but not too strong to move/affect other pieces around it.

### 4.3.4 Piece Movement

When it came to move the pieces on the physical board, there were elements that needed to be kept in place. All pieces had to be moved without collisions or changing the location of the other pieces in the board as it would affect the current locations and cause errors to the software that handled pieces location. The Movement system was either kept small so that the board doesn’t become unwieldy or utilized an algorithm that considered all the obstacles that were in place upon the board.

For the pieces to be able to move as indicated by the player, the system had to be interfaced with a microcontroller that connected with the XY board to simplify the number of possible moves that are available for the piece in place in any situation, and gave direction of starting and finishing positions.

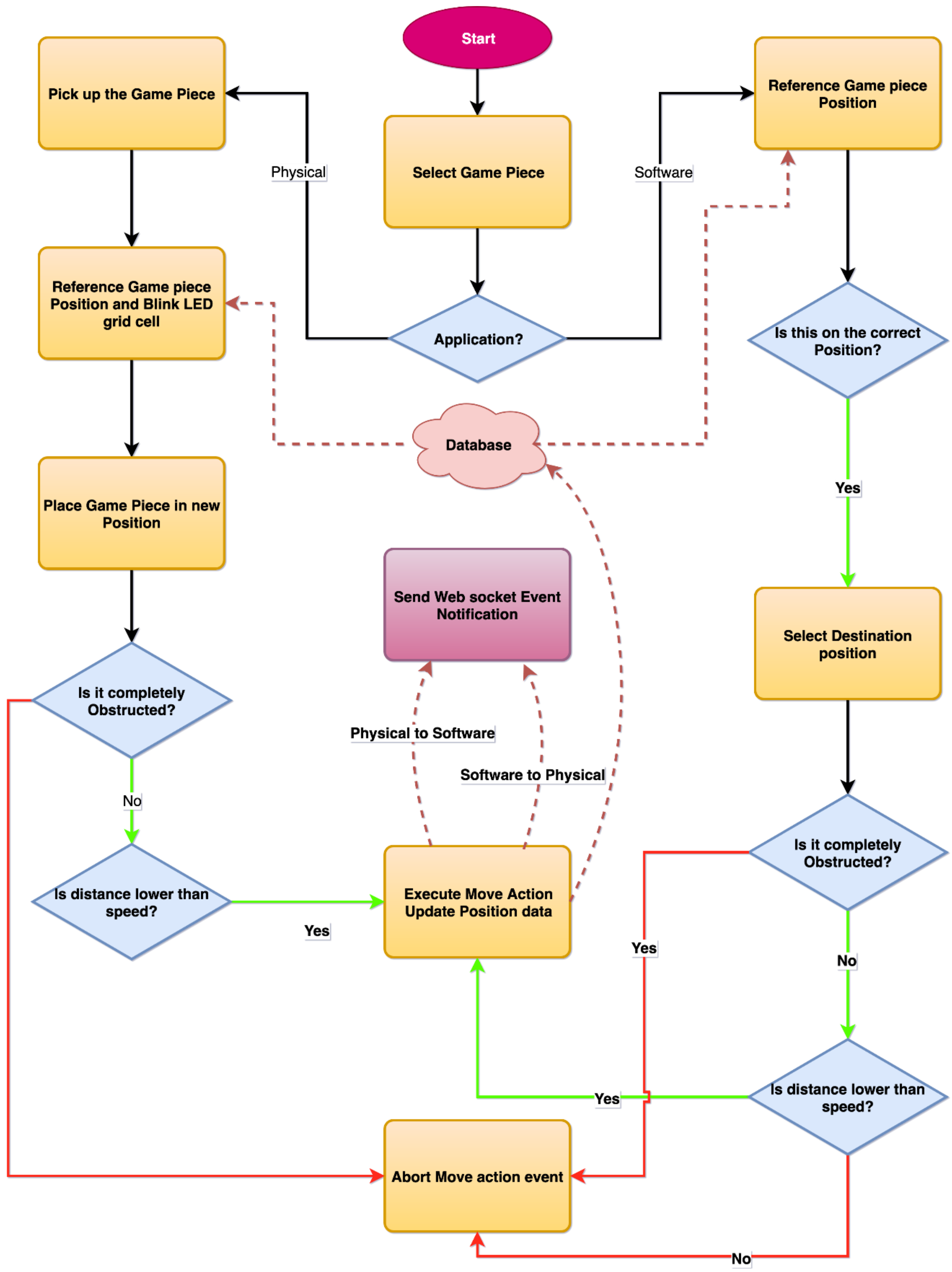


*Figure 8: Move Event Flow Chart*

Once the movement system was in place, it was able to work with the surface underneath the board itself. It also needed not to interfere with other systems that were in place such as the LED's and sensors. The goal was to make the movement mechanism to be hidden during play so that the game has an extra touch of realism and does not disturb or distract players.

When it came to executing the piece's movement, the flow chart within **Error! Reference source not found.** below displays the actions and requirements that will need to be done and fulfilled to make sure that the piece was repositioned correctly. Each action is considered a function or a physical action and is displayed at a high-level design. Each function stated within the flow chart that can be considered to have multiple actions needed will be explained thoroughly below.

For the Send Web Socket event, this had involved a communication API where the group established a socket connection with the web server and be assigned an ID per connected application. For this piece movement, the starting situation determined on when the event was sent. If the movement started from the physical board, an initial socket event was sent to the DM to verify if this a move or delete operation. This single action has operations involved with Section 4.3.5, since a physical move required a person to remove the game object off the board in the first place.

In the situation where the piece was slid around board to facilitate a movement, this can possibly trigger multiple sensors and at the same time give multiple event notifications. To combat this possible bug, clauses were put in to only record a single operation at a time and to add certain time limit to the final placement of a game piece. To explain, if a game piece was removed from its starting position and placed on a different grid location, the web application had to wait for a certain amount of time before finalizing that the position was the end point and then calculate the valid move event, which checked for obstructions and the amount of squares that were traveled with the pathing algorithm. Since there was only going to be a single move action that can occur on the application,

The full description of the Web Socket Communication is covered in Section 4.6.1.

When a game piece or token was selected to be moved across the grid, the type of event flow depends on what device the group was trying to move the object across. If the group was attempting to move the object across the tabletop application, it followed a certain set of events and then checked if this event was a correct action. This set of checks is matching how the web application dealt with piece movement, but the results of rolling back this action was done differently between the two.

Since all specifications have been established previously, the group discussed how the software was to set up for what has been discussed. The microcontroller received first the beginning and final coordinates of the piece that was going to be moved as well as the current coordinate of the XY board and the statues of the destination square. Once all this information has been received and processed by the micro controller it was now ready to begin processing all data received. First off, it started at looking at the destination statues, If the status returned 1, the software read and followed the full statues code and continued to the empty statues code. If the status was '0' then the software read and went directly to empty statues and skip the full statues code.

### 4.3.5 Piece Modification

Modifying the game pieces was implemented on both the web application and on the tabletop. If the DM added or deleted a token on the Web application, this added or removed the appropriate field within the database, in which the Tabletop application picked up as a change to the physical game board.

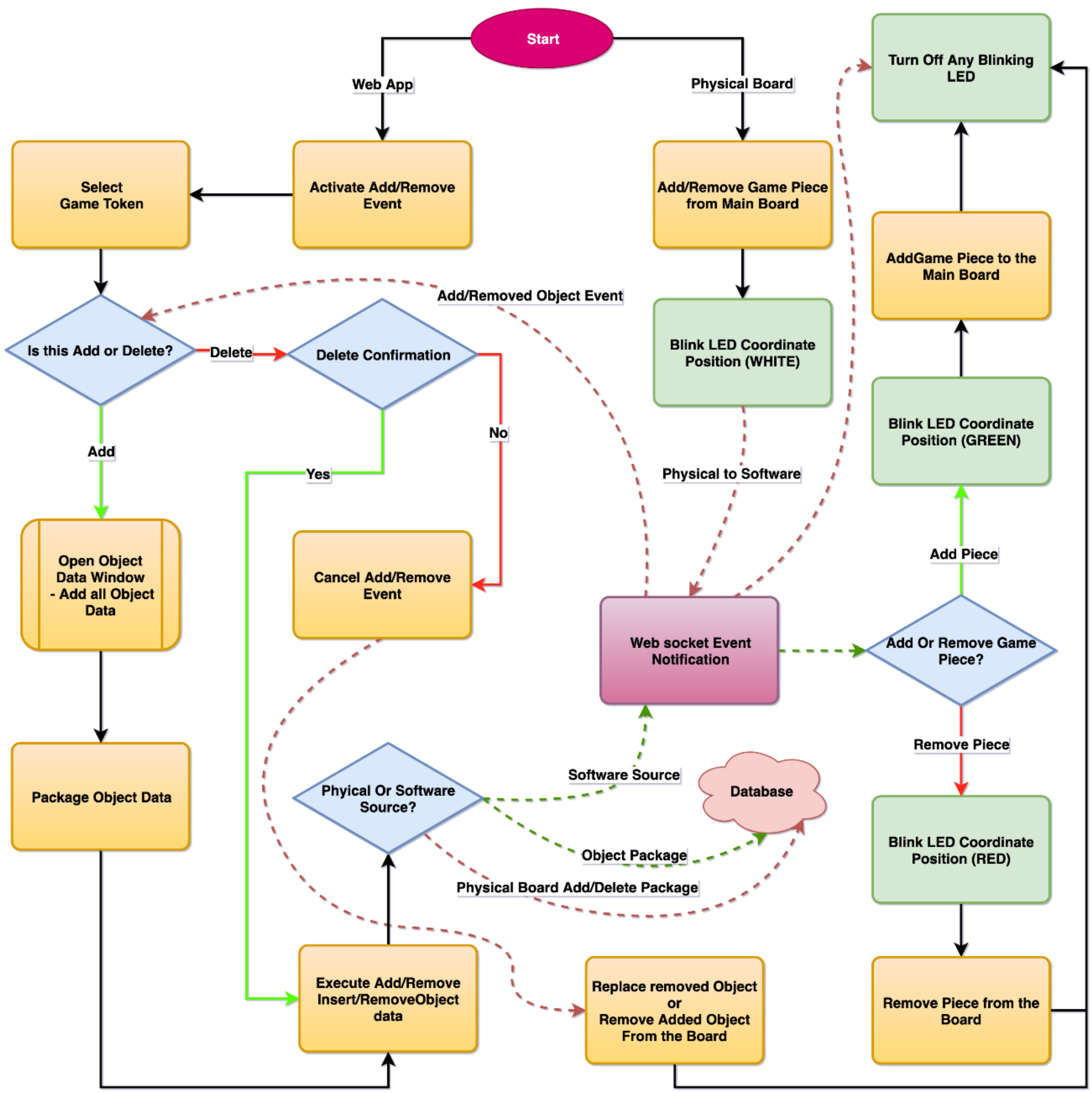


*Figure 9: Piece add/remove Tabletop LED procedure*

One solution to match the tabletop to the web app board was to color code the removal and add action. A green blinking LED symbolized an add action, while red blinking symbolizes the removal of a game piece. To see this represented implementation, refer to **Error! Reference source not found.**. Red Dotted lines indicate that the origin of the request was from the physical tabletop, while green dotted lines are sourced from the web application.

**Error! Reference source not found.** contains many high-level elements that reduced the number of events that were shown within this flow chart. At start, the action that was committed is done on either the web application or the physical tabletop application. For the web app, the user selected an object or space on the grid board and activated the add/remove functionality that displayed on the user interface. This displayed representation called back to the user interface, where the add and remove buttons was shown in a section that pops up at a game object selection. Activating either functionalities have put the application into a ready state for input data or deletion confirmation.

On adding an object, a display window popped up in the middle of the screen where the user can input the object data. Only the DM received this notification and was the only one who can input new data into the database. The different data types are shown in section 4.5.4, and was the required data needed for the game to progress smoothly without adding optional data types and fields. Once the data was inputted and was confirmed by pressing the finish button, then the data formatted and packaged into a json. This data package was inserted into proper table or collection, and assigned a unique Id, while a data insertion confirmation was sent back to the web server. From this point, the webserver sent a socket event to all connected clients on the WebSocket to update their grid positions by referencing the database. For the tabletop application, this socket event contained the addition info and coordinate position that needed to be updated. This was done by activating the LED position id with a green light and waited for a new reed switch activation on that position. Once the game piece was placed on the board and triggers the reed switch sensor, the new grid object position was completed, and a deactivation event was relayed to the LED to turn off.

Adding an object straight away on the tabletop application made the events flow in a different way, but ultimately ended up going through the same process of inputting information of the object on the DM's client. The process of adding the piece on the board on the final area was considered as a final check if the game piece was in the proper grid position.

Deleting a game object went through a shorter algorithm but required multiple checks to make sure that the action that was occurring was a final choice. On the software side, it moved straightaway into a confirmation window if the user was sure that he or she wanted to remove the game object and piece, while on the hardware side, it sent an event notification over the connected socket to the DM's web application for confirmation.

### 4.3.6 Path Planning

Each path the software creates was broken down into either the entrance/exit of the game piece and its displacement on the XY board. A main grid was used to set this up which is composed of two different grids called the minor and major grid. The minor axis was used to direct the entrance and exit portion and the major axis was used for the X/Y displacement.

The group started off the software by testing Xfinal and Xinitial with each other so that the group could see whether they were equal or if Xfinal was less than or equal to Xinitial. The same was done on the Y axis. Each combination between them gave up to eight possible outcomes and a set algorithm that allowed for a path to be created.

For the X/Y displacement, once we game piece was on the grid it could only move on the X and Y direction. For the exit & entrance, the game piece had only 4 possible directions in which it could enter or exit any of the squares on the board.

When a game piece was navigated on the board, the group anticipated obstacles and designated static grid spaces that were considered impassable. Moving a physical object on the board had the chance of possible collisions with other objects due to the XY plotter not being capable of considering the surface obstacles. So, a path plan was made at every move action that is considered final. This path plan was not implemented when doing physical movement actions from an exogenous event, such as a physically present player moving their game piece.

One possible solution for pathfinding was implementing a method such as Dijkstra's algorithm. It performs everything required, from finding the shortest path from the starting point to the goal, while taking in account obstacles that are in the way. It is also one of the simpler algorithms to implement, so that can help towards time considered in the development. The biggest flaw that Dijkstra's introduces is that in the worst-case scenario, in which there is multiple large obstacles and the goal position is far, the algorithm is very computationally expensive.

A solution to solve this worst-case scenario was to use a best-search-First method combined with a heuristic, such as the A* algorithm. It is a very efficient method to get a comparatively good path compared to Dijkstra, in a short period of time. This outcome is was found good for the application, where this algorithm had perform its calculations on the microcomputer. This algorithm could significantly improve the response time between different moves, but when it comes to implementation, the algorithm can be complex to program. A comparison chart between A* and Dijkstra's algorithm showcases the pros and cons between each method in Table 27 below.

| Table 27: A* and Dijkstra's Pros and Cons | | |
|---|---|---|
| | Dijkstra's | A* |
| ***Pros*** | • Finds the best path from entry to goal<br>• Easy implementation | • Fast algorithm<br>• |
| ***Cons*** | • Computationally intensive<br>• Exponentially slower | • Difficult to implement |

The planning algorithm chosen as the final choice is A* to calculate pathing. It had the speed advantage over its counterpart and less intensive in its operations over long distances. This algorithm is very popular in different applications and including a library that has this algorithm can make code implementation much easier.

Some issues that were initially addressed were if the group should allow the algorithm to include diagonal movement across the grid, how it would interact with the different game pieces that are currently on the board, and the possibility of it triggering multiple grid sensors at the same time. D&D allows diagonal movement which is a requirement that the application would have to follow. Solving this problem could involve adding a different type of movement that the XY plotter would have to execute, where instead of orthogonal movement, we would have the included diagonal movement.

Since the game piece path had eight directions to travel from a grid space, sensors were triggered that are not in the direct path of the game objects travel plan. This was solved while the XY plotter was in motion, triggered sensors during that duration were not tracked for any events. This assumed that the game piece arrived at its destination coordinate, while disregarding possible problems such as collisions.

When it came to collisions, the pathing plan took care of possible head on collisions. What it did not plan was for is possible indirect collision on game objects that were sticking out in various directions above the grid square. One way the group followed to take care of collisions was to follow the sensor plan for piece movement on section 4.3.4, where it went on keeping the tracking sensor detections but also had a delay on the destination. The plan was to add on top of that, where if any game objects near the path plan were suddenly detected on a different grid location, then they marked the original locations with lit LED's and ignored the new triggered sensors. Once those marked sensor locations were all triggered and after a small delay to determine the final movement, the tabletop application went back to being in a ready state. This flow is represented in Figure 10 below.

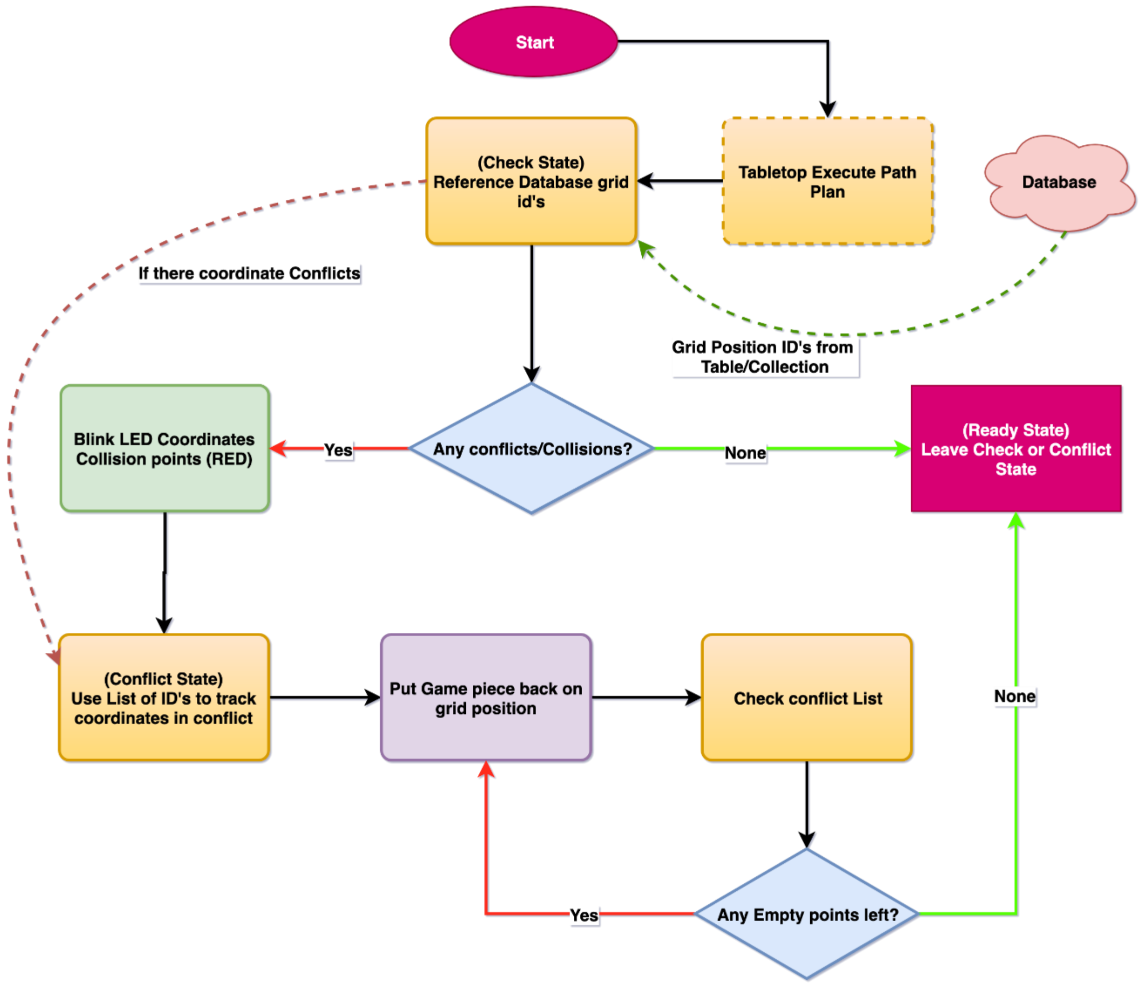


*Figure 10: Collision Conflict Correction*

## 4.4 Server Hosting

How the data was stored is important in making sure that all the assets and game objects were synced between the two applications. The main reason the group went towards web hosting was to allow the web app to be useable in all devices. This meant that the layout of the web application was key to making sure that all functionality was accessed with little difficulty. Pricing was also very important due budget constraints and there group was also not computing large data packets.

### 4.4.1 Hosting Platform

A Variety of platforms were researched for hosting the webserver and database. Since the requirements call for a real-time application, and preferences towards a JavaScript based backend, we looked at node JS as the webserver and MongoDB as the database. They also needed to access the server for custom installations and configurations, so for most hosting services, this was not allowed by default. The group also looked at having a solid web development stack so choosing a service that can support a solid package was preferable. Price was also a major factor that had to be accounted for as well.

The decision of going for a Platform as a Service (PaaS) or Function as a Service (FaaS) depended on the usage and the development of the application. FaaS architecture required a different methodology of designing the backend, so sticking with a more traditional architecture and host platform smoothened the initial development.

The first hosting service the group looked at was HostGator. It had its options at a very cheap price level and their services allowed for easy setup. Included into their service package is NPM, which allowed for various modules to be installed in a simple manner. The downside to using HostGator was that their pricing model for using these services was too expensive, sincea VPS package or a dedicated server package were needed.

Heroku was another hosting platform that was considered for the web app. It offered a free plan and is great for small projects. The free plan came with 512 MB of memory, with a given sub-domain. This free package also came with a few pitfalls, such as a single user access to the NodeJS server and 30-minute inactivity timeout. This restriction on users did not fair well with the team, since multiple people needed access and it would have slowed down development.

Another service that was noted is A2 Hosting. It has specific plan options for NodeJS hosting and its pricing is affordable. It integrates cPanel access, which would allow for simpler navigation of the server elements. One of the biggest perks of using A2 Hosting is that all their packages contain unlimited storage and bandwidth. Its great for future expansion if further development is made on the application.

NodeChef was specifically for NodeJS and utilizes Docker containers. It showcases its dynamic system to using their services and scaling up resources when needed. One great aspect of Docker is that it allows for an easier learning curve for its setup and is great for development teams. One example is its straightforward method of having its local environment closely matching to the production environment, so that integrating the newly developed container full of feature changes will be as smooth as possible.

AWS (Amazon Web Services) was another direction that the project could go for hosting. The amount of resources and support is top-end and the multiple options for Node JS setup, such as using Amazon’s Elastic Compute Cloud (Amazon EC2). Another option was to use Amazon Amplify, with both choices helping in deploying the application faster, with the 1st choice more suited for large data analysis and processing. Amazon Amplify provided a full work flow for git development and full stack server-less applications.

One other hosting service that has been factored for was Microsoft Azure. Its powerful features allow NodeJS deployment to simple through containers such as Docker or Kubernetes. It also integrated Visual Studio Code within its build process and extends the development process with built-in git processes. It fits all the requirements except for pricing, which is priced by hour or by month, and is to expensive for the budget.

The last service that was considered is RedHat Openshift. It allows for three simultaneous processes within Node JS and it is part of its free plan. This also includes 2GB of memory, so for the usage it is considered fair since we are not dealing with heavy data manipulation. The downside is the 30-minute inactivity timer, as well as having a Maximum active usage timer within a 3-day period. Gaming sessions within D&D can easily reach 6 or more hours of usage per session and can happen at any time. This would mean this service would be more fit for long downtime sessions and will not be able to accommodate multiple sessions per day.

Ultimately, due to time constrains no hosting platform was selected for the software integration of our project.

## 4.4.2 Web Server/Stack Selection

Node JS was the chosen webserver to utilize a single page web application. This decision was done after analyzing the functionality of using an apache-based web server, and its capability in solving the issues that were involved with the project. It was fully capable of being the back-end that supported a dynamic and modular web application, but to utilize that full functionality required a lot of complex programming and 3rd person libraries and API's.

The web developers who worked on the backend had to integrate PHP, JavaScript, and HTML in a cohesive manner without causing bugs or extra issues. This extra overhead introduced more development time, so the choice of sticking NodeJS eliminated one programming language the group would have had to learn. The following table displays different web stacks which included the advantages and the disadvantages in terms of learning requirements, different functions from web stacks.

| Table 28: Web Stack Comparison Table | |
|---|---|
| **LAMP** | – LINUX, APACHE, MySQL, and PHP.<br>– open-source coding framework and free for users.<br>– Multiple languages to learn can increase learning curve. |
| **MEAN** | – MongoDB, ExpressJS, AngularJs, NodeJs.<br>– Popular with various computer science and engineering fields.<br>– Relatively new software stack<br>– Smaller community of developers could mean less support. |
| **METEOR** | – Possible second choice other than MEAN.<br>– Contains NodeJs and MongoDB database.<br>– Generally used for smaller applications |
| **RUBY ON RAILS** | – Easy learning curve compared to other stacks<br>– Due to its ease, there is less control on certain functionalities.<br>– Could lead to code quality issues if not managed correctly. |

For the web stack LAMP, it consisted of LINUX, APACHE, MySQL, and PHP. This is a free web stack that contains open-source coding framework and free for users. However, this web stack was found difficult to use due to multiple coding languages being unrelated.

MEAN stack offered MongoDB, ExpressJS, AngularJs, NodeJs. This web stack can be a favorable choice since computer engineering students have knowledge of NodeJs. This web stack is relatively new in terms of software development which means that it contains lack of community of users and/or developers that can troubleshoot any problem that may arise in the project.

METEOR web stack was the second choice the group selected. It contained NodeJs alongside with MongoDB database which made the coding framework easier. However, this stack was meant to run small applications which it would have been a problem for this project.

Ruby on Rails was a favorable choice for many novice coders and developers which was ideal for the project, however, if offered no control of the web application. This web stack ignored coding fundamentals and could potentially crash the web application and lead to long debugging sessions.

Ultimately. A variable on the Mean Stack was chosen for our project. It used NodeJS as our webserver but implemented a new version of angularJS called Angular and also instead of using MongoDB a local database store using JSON files was selected instead.

### 4.4.3 Database Selection

Mongo DB and MySQL were the two main databases considered for this application. They both have their pros and cons when it came to usage and integration. So, to show this comparison in a straightforward manner, we compared the main subjects that were related to the project within a clean table. This representation is shown below in Table 29.

***Table 29***: ***MongoDB vs MySQL Database***

| | MongoDB | MySQL |
|---|---|---|
| ***Pros*** | • Well Documented<br>• Fast Data retrieval<br>• Realtime transactions<br>• Dynamic Schema<br>• Single Language<br>• Direct usage of JSON | • Classic/Legacy<br>• Well Documented<br>• Structure Relational Data<br>• Relational-base Tables<br>• Good for large datasets |
| ***Cons*** | • No Joins<br>• Not relational | • Schema is usually static<br>• Requires knowledge on SQL<br>• Conversion process to JSON |

Choosing between MongoDB and MySQL depended on what was easier to develop vs its capabilities in meeting the requirements and more. NodeJS employed either database and each had a fair number of pros and cons against each other. Going for MongoDB would have the benefit of only using JavaScript as the backend language and, for the most part, the front-end as well.

Ultimately, when group implemented a local database over MongoDB due to time constrains, internet connection issues as well as budget issues.

## 4.5 Database Design

The database design was important for making sure that the data tables or collections were optimized, efficient, maintainable and that the design was also scalable. When it came to the long term, the schema was expected to evolve to include other elements, so MongoDB was the excepted choice. But when it came to the short-term goal, a certain set of data was expected to be the same at this time so MySQL could also be considered for this application. Whatever the choice, the schema could be adapted to use the chosen database to its best potential.

### 4.5.1 Database Schema

The schema contained multiple sets of data for various types of functionalities. It was separated into major sections, all of which were interconnected in some way. MySQL uses a relationship setup, while MongoDB uses Document oriented connections, such as linking. This emulated the same process of having a one-to-many relationship and being able to normalize the various collections. There are also embedding a sub-document to parent documents, which helps towards reducing read and write calls, but can have the side effect of increasing the document size and data repetition. Both methods were ways to structure the data, and to improve the throughput, embedding sub-documents will likely be the most utilized method within the database.

Even though MongoDB has a schema-free structure to its database, it was important to have a schema design to the collections and documents so that the data can be retrieved in easy manner.

The overall design looks different between these two schemas, but the goal was to separate the data into their own categories. These categories included User data, Game Token/Piece Data, and Hardware data, and were covered in the following sections in full detail. The initial MongoDB schema was represented in Figure 11 as shown below. Every collection that has some sort of connection to another is referenced within their collection.

The move list collection is a special set of documents that will be used to record the coordinates for the move action.

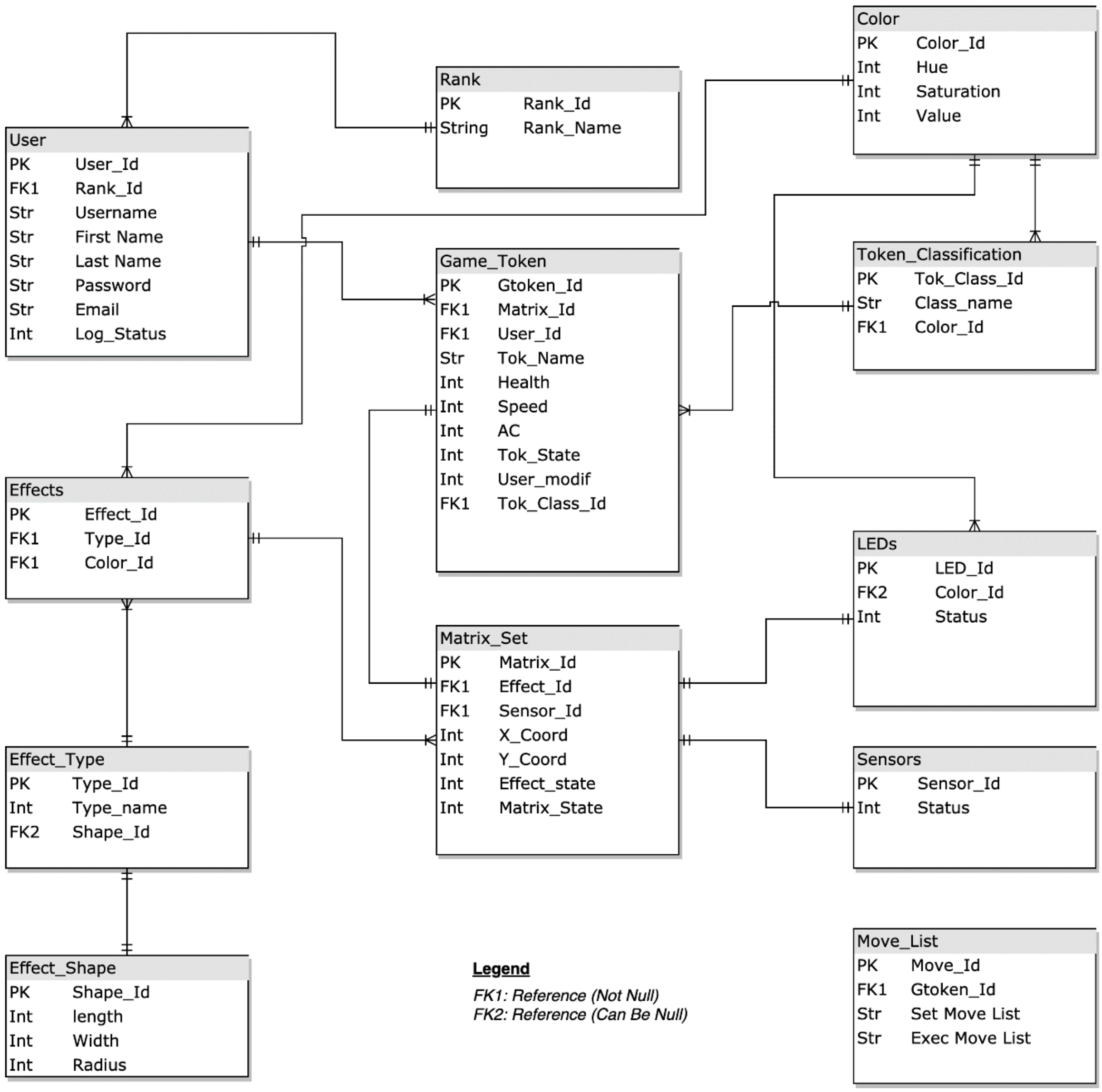


*Figure 11: Initial MongoDB Schema*

## 4.5.2 User Data

The data that was applied to users will need personal information for logging in and keeping track of their logged in status. This information covers the username, first and last name, and an optional email. Leaving email as optional will be a personal choice for the user, while the username, first, and last name are mandatory. Each user will have primary key User ID for their account, as well as a rank ID. Ranks are a sub-document to the User Document collection, and will contain the rand ID, and name for that rank, which was separated into “Player”, “Dungeon Master”, and “Administrator”, with each following rank having more privileges than the last.

Ranks were important because they represent what actions a user can perform while using the web application. A user with the Player rank can only control the specified game tokens that an Admin or Game Master assigns to them. The limit of controllable game tokens is set by either of those roles, and technically this means a user could have numerous tokens under their control, but this number is controlled by the amount of available spaces that are left on the tabletop. For a 16 x 16 board, there was a limit of 256 different tokens that can be properly placed on a grid space, not counting restricted grids.

### 4.5.3 User Security

Storing passwords as a plain string field for users was not a valid solution, so to ensure user security, hash/salting a password will be required for all user entries. Bcrypt.js is a JavaScript NPM library that has been considered for hashing passwords and most likely will be the choice for establishing the website security. Other security plans have been considered, but a free option was the main objective to reduce costs.

A Third-party software such as Twilio's 2-Step Authentication API has also been considered to improve the security. It would drastically increase the security of the website and has a well-documented procedure to implementing the feature on various types of website. This would also involve having to sign-up for a free plan, but this would limit the user count and the authentications are set to less than 100 per month. Work-arounds can be implemented to circumvent this limitation but further research would have to go into setting up an extra user interface to confirm trusted devices for the selected user account.

### 4.5.4 Game Piece and Token Data

Token Data contains the unique ID associated with the token, Reference ID's and its basic information. This includes the name, health, speed, and armor class (AC), its current game state, and its coordinates. The reference ID links to the User and Matrix ID that it is associated with. This data represents a single entity across all the connected clients and the tabletop application. As such, every token data was unique and must be on a specific location on the 16 x 16 matrix.

When a change was executed on a token on the web app or the matching game piece on the tabletop, its state was updated to the execution state and the User_modif field was updated with the data representation of the User ID. This was part of data synchronization and will insure that only one person can make changes to a specific game piece at a time, and to prevent conflicts between user actions.

## 4.5.5 Hardware Data

Hardware data is all the changes that are coming in from the Tabletop application. Sensor and LED data must always be synchronized and when it comes to the overall session, since the tabletop is the most used device between all connected clients. Each sensor and LED have a unique ID associated with it and their respective collections are at a fixed size. They are linked or sub-documented to the Matrix Set collection, which in turn has a unique ID with a fixed coordinate set.

The matrix set was set to 256 rows or in MongoDB, documents, of data. This equates to the 16 by 16 grid that was the main playing area for all the users within the session. This 256 data limitation was set in place due to budget constraints, since increasing the tabletop space introduces large increases of sensor and LED components. When it comes to the software back-end, this grid space can be easily scaled up due to the ease adding more documents to the matrix set collection. To represent the extra cost due to physical hardware, Table 30 is displayed below, to show the overall cost increasing the tabletop space, while Figure 12 shows a visual of the increase of price vs the increase of grid space. A single component was estimated to be $0.40 each.

***Table 30: Grid Size vs Cost***

| Grid Length | # of LED | # of Sensors | # of Components | ($) Overall Cost |
|---|---|---|---|---|
| *2* | 4 | 4 | 8 | 3.20 |
| *3* | 9 | 9 | 18 | 7.20 |
| *4* | 16 | 16 | 32 | 12.80 |
| *5* | 25 | 25 | 50 | 20.00 |
| *6* | 36 | 36 | 72 | 28.80 |
| *7* | 49 | 49 | 98 | 39.20 |
| *8* | 64 | 64 | 128 | 51.20 |
| *9* | 81 | 81 | 162 | 64.80 |
| *10* | 100 | 100 | 200 | 80.00 |
| *11* | 121 | 121 | 242 | 96.80 |
| *12* | 144 | 144 | 288 | 115.20 |
| *13* | 169 | 169 | 338 | 135.20 |
| *14* | 196 | 196 | 392 | 156.80 |
| *15* | 225 | 225 | 450 | 180.00 |
| *16* | 256 | 256 | 512 | 204.80 |
| *17* | 289 | 289 | 578 | 231.20 |
| *18* | 324 | 324 | 648 | 259.20 |
| *19* | 361 | 361 | 722 | 288.80 |
| *20* | 400 | 400 | 800 | 320.00 |
| *21* | 441 | 441 | 882 | 352.80 |
| *22* | 484 | 484 | 968 | 387.20 |
| *23* | 529 | 529 | 1058 | 423.20 |
| *24* | 576 | 576 | 1152 | 460.80 |

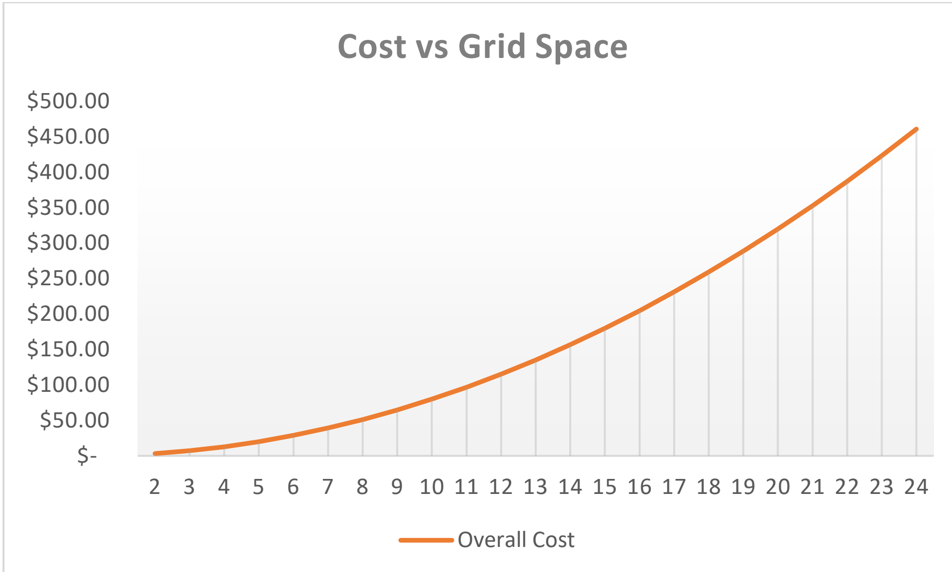


*Figure 12: Cost Vs Grid Space*

## 4.6 Data Synchronization

Data Synchronization was extremely important to making sure that every web client and device was synced correctly and utilizing the same data set. There should be no discrepancy with the data that was used with all the connected clients and the physical board application. That means that a peer-to-peer connection was a poor solution to making sure that all the data on each client was matched correctly to every other client. The direction that is preferred was a client-Server, where the tabletop application was connected to the server, while the web app clients are doing the same.

To facilitate this equilibrium between all the devices, the group had an event and messaging system for any action that was taken on the web apps, as well as on the tabletop app. This does not include the sensor loop algorithm that was required for the tabletop app. This was also buffered by the state system that makes sure that when a game piece was being executed with an event, that piece can only be modified at by one person at that time.

Data communication will ensure that for the project, that regardless of data modifications on local application, all changes are merged with the original data source.

### 4.6.1 Web Socket Communication

Socket Communication was key to making sure that every client is updated with the current state of the server. This implementation would involve a user activating a functionality within a single client and this client would send a web socket event to the server, and then a server message to all connected clients.

Web sockets are useful within this project because it allowed for asynchronous real-time communication. This also makes it valuable for the single page web App, since all the functionality must run continuously.

In order to communicate using the WebSocket protocol, we needed to create a WebSocket object; this will automatically set up a connection to the server. Some protocols that are needed can be put into a string or an array. These strings allowed us to indicate sub-protocols, so that we can implement multiple WebSocket sub-protocols. This allowed the server to be able to handle many different types of interactions.

### 4.6.2 Database Validation

Data that is inserted into the database needs to be valid, since faulty data can lead to errors that could be harder to track. These are harder to track since the database collections are expansive and can possibly include hundreds of entries throughout the course of a couple of game sessions. There is already a set number of entries for the sensor, LED, and matrix set collections and if invalid data is set within any of those collections, it would automatically cause an error on the first loop of a game session cycle.

There are numbers of validation types we can do to check the data that is being entered. The following validation types Table 31 and their explanation are displayed below.

| Table 31:Database Validation Types | | |
|---|---|---|
| **Validation Type** | **How it works** | **Example usage** |
| Check digit | The last one or two digits in a code are used to check if the other digits are correct | Bar code readers in clothes use check digits |
| Format check | Checks data is in the correct format. | Check digit is computed from the characters given and compared to the given check digit |
| Length check | Checks data is not too long or too short | A password that needs a certain amount of letters long |
| Lookup table | Looks up acceptable values in a table | There are only 24 hours in a day |
| Presence check | Checks data has been entered into a field | In many databases a key field can't be left black |
| Range check | Checks that a value falls within the desired range | Number of hours studied for a test must be between 5 and 15 |
| Spell check | Uses dictionary to look up words | Word processing |

## 4.7 Communication

For both the physical tabletop board and the web app VTT to work towards the specifications, the communication between both applications needed to be in sync.

Also, any actions used to communicate between an offsite player and the rest of their group must be user friendly and recognizable. The Database was main repository for all action and object data. It also serves as verification for any events that can happen on both applications.

### 4.7.1 Server to Application Communication

Androids app can communicate differently to the server in many ways. Any server-side technologies like JSP, Servelts or PHP are able to provide services and Apps can consume them via Rest API. Below Figure 13 represents how the format would look

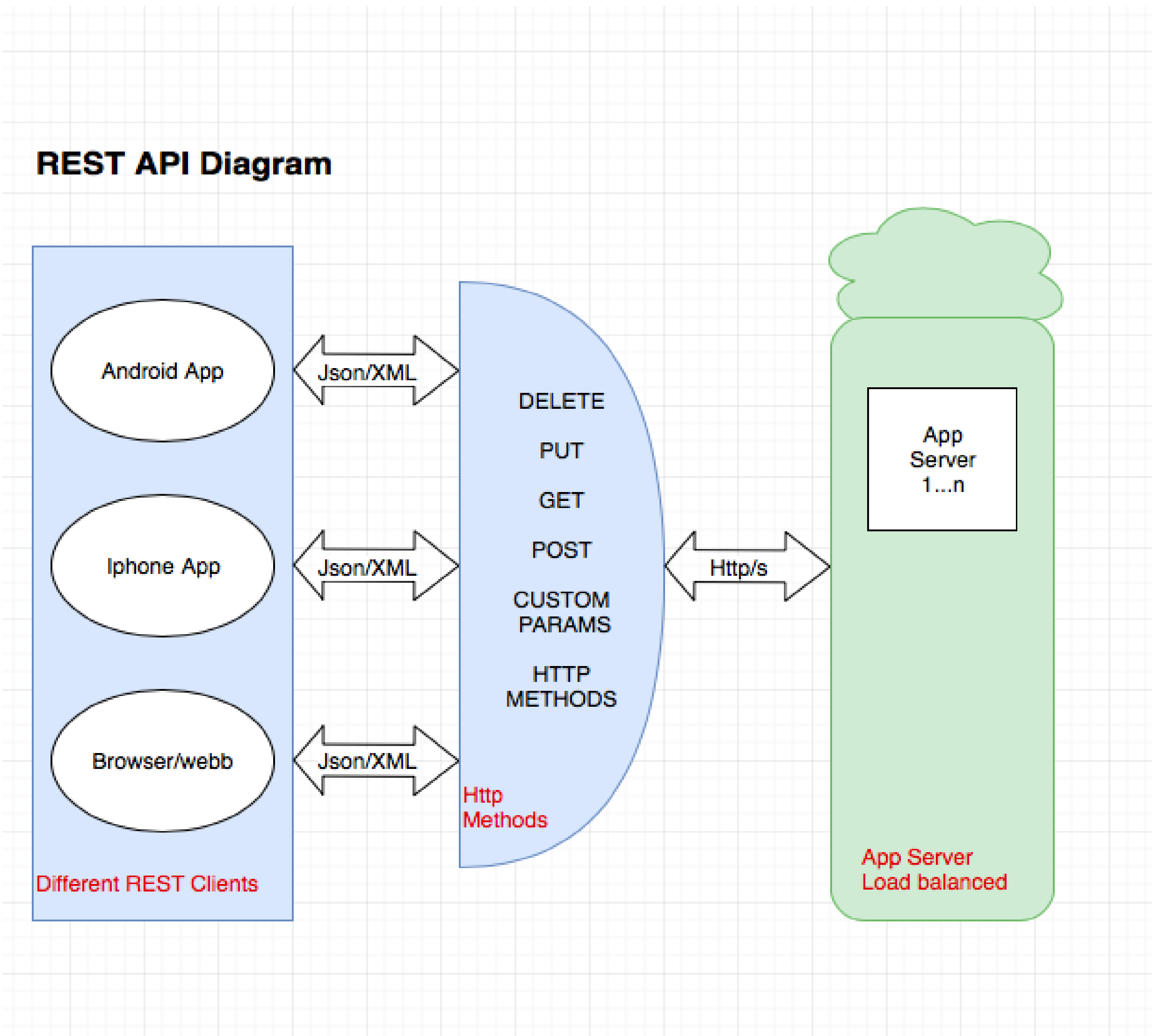


*Figure 13: Rest API Diagram*

The purpose of the REST API is to break down a transaction to create small modules, each module addresses a particular underlying part. This type of modularity gives us a lot of flexibility but for beginners may be challenging to start from scratch. As explained above, the group used HTTP request to GET, PUT, POST and delete data. GET was used to retrieve a resource; PUT was used to change the state or update a resource, this can be a block, file or object. When doing POST, the group created the source and lastly DELETE was done to remove it.

For using a RESTful API, there were some constraints that needed to be evaluated. The first one was that it is Client-Server based. This means the group needed to make sure the client and the server are separated from each other and allowed to evolve individually. The group cannot mix UI and request-gathering which are the client’s domain with Data access, workload management and security which are the server. Otherwise potential conflicts can come from it.

Since RET APIs are stateless, calls can be made independently of one another, and each call needs to contain all the data needed to complete itself successfully. The resources should catching, because a stateless API can increase the requests overheard by handling many large amounts of incoming and outbound calls.

One disadvantage of using REST APIs is it can lose the ability to maintain state in REST under the sessions created. The members of this group are all new developers, so REST could result challenging due to their latency in request processing times and bandwidth usage. Overall, because of its great flexibility, lower maintenance costs, high scalability and simplicity the group used REST API for server to web app communication.

### 4.7.2 API Utilization

To facilitate the communication between both the physical and virtual devices, a variety of API’s was used in order to streamline the development process and to reduce workload. These API’s include socket event communication and database communication.

### 4.7.3 LED Utilization

Utilizing the LED’s was straight forward using the FastLED library. This library can be loaded into the Arduino IDE and be used to control each led on the 16 by 16 matrix. This library supports the favored LED design that  are expecting to use on the project, which is a WS2812B addressable LEDs. Their simplicity and broad utility allowed to light up the board with various patterns and gives a tool for the web app users to interact with board and communicate more efficiently. The overall LED design is covered in section 5.2 and other considered LED products.

The functionalities that the LED's will display were controlled by the web app users. These include pinging, moving, and action events such as spells. The set of colors the group used for the LED's was stored on the database and can be referenced to multiple LED's with a call function. Pinging is an action where a user wants to call attention to a single spot. This can be a simple utilization of calling the LED's array position with a color and blinking the led with a delay function.

Moving a user will also utilize the LED's to highlight the pathing of a game piece moving across a board. The initial starting point was marked with a blue lit LED, and the path traveling to the destination point was marked green. The final grid position was marked with a red LED and turns off once the game piece has traveled its path to its goal. Possible collisions have been covered in 4.3.6, and the collision grid spaces will be marked with red flashing LED's.

The most challenging LED algorithms was the action and spell events. Certain actions can be singular LED actions, but the spells include complex shapes that can be a challenge to implement correctly. There are five different type of shapes a spell can take. These include a cone, cube, cylinder, line, and square shapes. If these shapes were set, the group would be able to just store a set amount of prefixed shapes per type of spell but many of these spells can be directed in in nearly any direction. To simplify the dimensions, the group is leaving out the three-dimensional aspects of the spell and restricting the effects to two-dimensional space.

To calculate the effected grid spaces, a line was drawn from the origin grid space and projected out to the designated distance of the spell. Every spell in DND was counted in increments of 5 feet, in which the grid spaces are 5 feet in diameter within the game. From there, the type of spell was calculated, and the algorithm will determine the grid spaces are that are affected. The visual representation of these spell effects is shown in Figure 14 below.

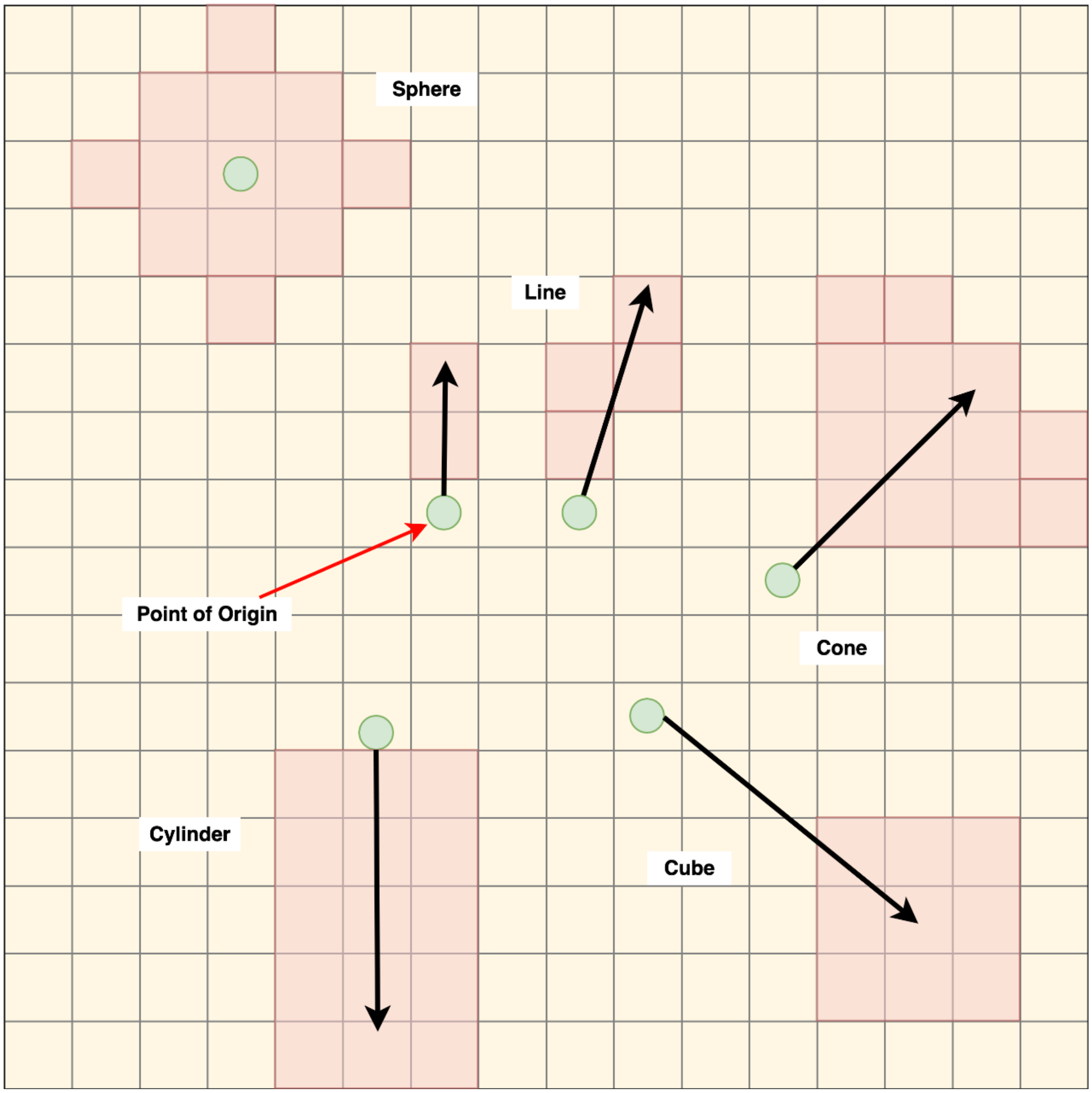


*Figure 14: Spell Area of Effects on a Grid*

The spells that can be pointed in a direction needed to be drawn out from within the web tool and their corresponding spell effect will be applied based on what type of spell is being used.

### 4.7.4 Microcontroller and/or Microcomputer

There will be a few microcontrollers considered for this project. The MSP430, Arduino Mega 2560 and Raspberry Pi. The MSP430 is a 16-bit microcontroller that has a number of special features. It is known to be extremely low power consumption having only 4.2 nWper instruction while also having high speed 300 ns per instruction hence why used in many projects. Arduino Mega 2560 is designed for more complex power as it has 54 digital I/O and 16 analog inputs which makes it more suitable for 3D printers and robotics projects.

The Raspberry PI is also a low cost, credit card sized computer that has the capability of doing everything expected in a desktop computer. It has the capability to interact with the outside world and has been used for digital maker projects such as music machines to parent detectors to weather stations.

The group has done extensive research for each micro-controller mentioned and when it came to price, availability of part, and easiness of use, it was determined that the Arduino Mega 2560 would be the best suitable microcontroller. What makes this micro controller very suitable for the project is that on top of being low cost, low power consumption and high speed it also has the capability of having wireless LAN and Bluetooth connectivity which allows it to connect with the other components such as the sensors, LED’s and X/Y plotter.

The Arduino Mega 2560 has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs, a USB connection, power jack, an ICSP heater and a reset button. It has everything needed to support the microcontroller. The group needed to connect it to either a computer with a USB cable or power it with a AC-to-DC adapter to get started. For the sensors used in the project, their outputs were fed into the microcontroller that was being used. They represent the equivalent electrical values of any physical quantity. Sensors generally measure physical quantities like distance, light intensity and acceleration. When interfacing with the microcontroller, the microcontroller’s digital input pins will directly accept the sensors digital output. Once a good communication has been established between the sensors and microcontroller, the group was able to send signals to the output devices which were the X/Y plotter and LEDs in order for the tabletop game to work as expected. Below is Figure 15, a detailed graph explaining the connection.

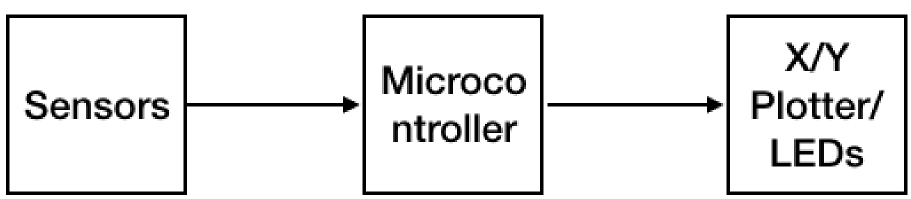


*Figure 15: Sensor Diagram*

Many interface methods were considered by the group in order to solve the complex problem of balancing circuit designs criteria such as cost, features, size, power, weight consumption, manufacturability, etc.

For the sensors output signal, the group has considered some options that would fit better for the project criteria's. Some of these options include having an analog voltage output or digital output. When it comes to Analog output, they are voltage-based control and monitoring, their voltage currently ranges from 0 to 5V with a current range from 0 to 20mA.

Following an analog output has some advantages as well as disadvantages. The main advantage the group found beneficial to the project is that is has a simple interface process. This was very beneficial since there are many complications that can come from the interface such as when the sensor's output-voltage range is much smaller than the ADC's input-voltage which results in a drop in resolution, having a tight error budget, using a thermistor but having a linear temperature-to-code transfer function, and having limited ADC inputs. When the interface process is simple, all these complications are greatly reduced. Other advantages of using an analog output include low programming overhead, high speed and low cost for low resolutions.

Although some of these advantages are great, it also comes with some disadvantages, the main one the group found was having a high cost for higher resolutions. Since the microcontroller has 40 inputs this can result in a very pricey output connection and just a high cost project all together. Other disadvantages are it complicates the circuit design when external ADC or DAC are needed, has a very short distance (only few feet maximum) and lastly not all microcontrollers have analog inputs/outputs built in.

Digital outputs connect to digital input ports and only communicate through a serial protocol. These sensors are auto-detected by the climate monitor, the signal measured is converted into digital signal output inside the sensor and then it is transmitted through cable digitally. Lately, they have been replacing analog sensors as they can overcome the drawbacks that analog sensors have. Just like with analog outputs, there are advantages and disadvantages of using digital outputs. The main advantages were that they have higher performance, higher reliability and are easier to design and maintain. This is very important for the project because for most people in the group this is their first big project and having an interface method that is easier to design and maintain can result in less time and money spent on it for tight error budgets.

The drawbacks of using digital signals are that it requires data conversion and if one is not familiar with converting data it can deem difficult. It also cannot provide continuous output for every change in input parameters which analog sensors can. Lastly, it costs more than analog. Below we have Figure 16, which is a graph explaining the interface with analog/digital sensors with the microcontroller using a temperature sensor as an example.

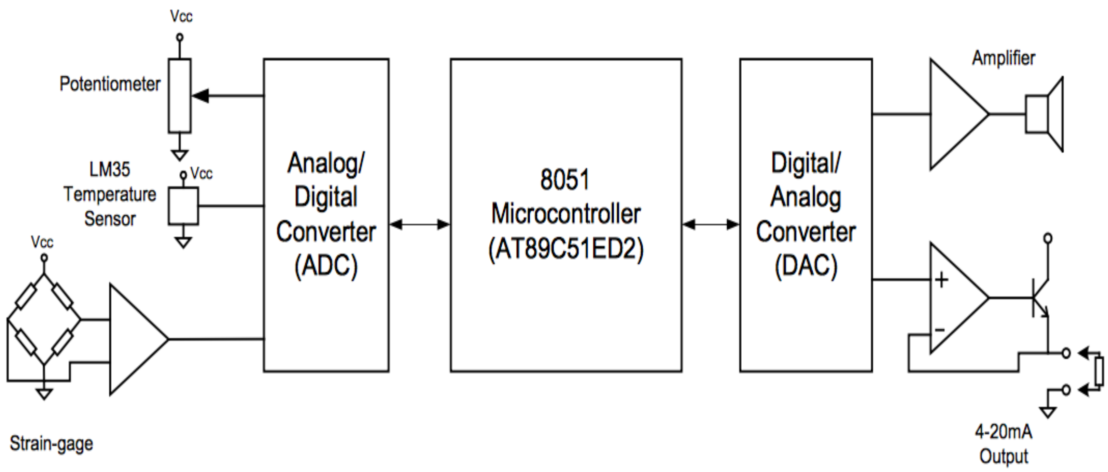


*Figure 16: Analog Input/Output*

## 4.8 Code Library Usage

A software library is a suite of data and programming code used to develop software programs and applications. It allowed us to assist the programmers working on this project and the programming language compiler in building and executing software.

The group expects to find pre-written code, classes, procedures, scripts and more to benefit for. Also, in order to reuse code for the project there is some guidelines that needed to be followed

There are restrictions in using code libraries because of the easiness it is to get them. Since a programming language is somebody's intellectual property, all code is therefore inherently copyrighted, and the group needed to treat it as such. The code obtained is no different from the text in a book, so for us to use the code, the group would have to pay the original author a listening fee. Using code without regard to the creators' right is both illegal and unethical and could lead to potential legal trouble.

As mentioned previously, fortunately the team can obtain rights to use a code library by obtaining a license. One license obtainable is a creative commons license. This dictates whether the code that is wished to borrow can be used for commercial applications or strictly for non-commercial applications. It also lets us modify the code the group has borrowed; this is important since most code written is applicable to the owner's project. By modifying it, it can use it to the project or any project desired

Even with a code license, abiding by all terms can become tricky. For us to avoid any legal problems, the group needed to make sure license limitations were met. A good plan that can be developed is to submit code libraries for review as they are introduced into the project, this practice would allow us to potentially cut weeks or even months from the project development schedule. The IPO can start the process of inspecting the code libraries while the software or application is being developed, a complex license could take up to month or more to review so it is important for us to request a license review as soon as they start designing the project.

# 5.0 Hardware Design

The hardware design in this project consists of the sensors, LED's, LED drivers, FET multiplexer/demultiplexer, and the XY plotter. This chapter covers the different types of sensors, LED's, LED drivers, FET multiplexer/demultiplexers and their alternatives that are used in the project. With the XY plotter there are many things that go into it like the stepper motors and the electromagnet that it will have. Each of these hardware parts are critical to get the project working and for it to perform its physical functions for the game. Keeping all this in mind it is also critical that the hardware that is chosen in the project can also be compatible with the software parts because they need to be able to be programmed to do what is necessary for the game.

## 5.1 Sensors

Sensors are important and needed for the project to serve the purpose to detect the location of where a game piece is for the software to read it. They are an important part because they will detect the XY plotter's electromagnet and will react to it therefore moving the game pieces around. Research was done to find out what type of sensors would be needed to apply and work in the project.

Since they need to react to the XY plotter's electromagnet that means they must be magnetic sensors. In this section different types of magnetic sensors are discussed and compared to show which one would be best to use for the purpose of the project. The two found to work best for the purpose of the project are reed switches and hall-effect sensors. There's also another magnetic sensor that would work well in the project called a magnasphere but is not necessary. Now, to explain them more in detail.

### 5.1.1 Reed Switches

Reed switches are one of the two types of sensors that are strongly being used and will most likely be used for the project. To explain what reed switches are and how they apply to the project, reed switches have two ferromagnetic blades, when a magnet gets near the two blades will close and touch when normally the blades are open and allow the electricity to flow. Then, there are reed switches made from non-ferromagnetic material and the difference is instead of closing when a magnet gets near it to let the electricity flow it opens and the blades separate from each other instead to let the electricity flow. Either a ferromagnetic material or non-ferromagnetic material reed switch would work for the project. They also do not rely on a power source or need to be wired to anything to work. A reed switch, with it being magnetic will connect with the XY plotter's electromagnet and to detect the location of the game pieces.

### 5.1.2 Hall-Effect Sensors

Hall-effect sensors are the other type of sensors that were strongly considered being used in the project, but most likely will not be. To explain what hall-effect sensors are and how they apply to the project: first, hall-effect sensors are magnetic, which is what they need to be for the project, so they can connect with the XY plotter's electromagnet and detect the location of the game pieces. They are semiconductors and depend on a power source to work. There are two types of hall effect sensors, the two types are latched and non-latched. For the project, latched hall effect sensors would be the ones preferably used. This is represented in Table 32 below.

**Table 32: Latched vs. Non-Latched Comparison Table**

| | Latched | Non-Latched |
|---|---|---|
| **Reaction to Poles** | – Hall-effect sensor detects and reacts when the north pole of a magnet is near it. | – Hall-effect sensor detects and reacts when the north pole of a magnet is near it. |
| **Magnet Activation** | – Stays activated even when the magnet is removed. | – Only stays active when the magnet is near it. |
| **Magnet Deactivation** | – Only turns off when the south pole is introduced. | – Turns off when the magnet is removed. |

As shown, latched hall-effect sensors would be the best option for the project if it were to be used because it is desired to keep the sensor active even when the XY plotter's magnet is removed. The sensor is desired to be kept active, so the game pieces stay in place magnetized to the board and so they are not easily knocked off.

### 5.1.3 Reed Switches vs. Hall-Effect Sensors

As explained, both reed switches and hall-effect sensors can be used for the project. They both will perform the same function that is needed for the project to connect with the XY plotter's electromagnet to detect the location of the game pieces. Making a comparison of both reed switches and hall-effect sensors shows that price-wise they are both similar and it's not a significant factor to choosing one over the other, but the research shows that reed switches significantly will work better and are more efficient for the purpose that they are needed for the project.

Also, even though the sensors themselves are similar in price a hall-effect sensor would turn out to be more expensive because of the power and circuitry they need to work. The Table 33 below shows the comparison of both.

**Table 33**: **Reed and Hall-Effect Comparison Table**

| | Reed Switches | Hall-Effect Sensors |
|---|---|---|
| **Resistance** | – They do not face as much electrical resistance because of the closed contacts that they have.<br>– Can be measured to be as low as milli-ohms in resistance. | – They have a lot more electrical resistance. Can measure in the hundreds of ohms. |
| **Efficiency** | – They work with a variety of frequencies, voltages, and load. | – Very limited with the amount of frequencies, voltages, and load that it can work with. |
| **Connectivity** | – They work whether they are connected or disconnected to a wire and don't need a power supply. | – They need to be wired and require circuitry to even work. |
| **Durability** | – They are durable and withstand a three-foot drop without being damaged. | – Not as durable and can be damaged easily. |

As shown above, reed switches are undeniably the best and most cost-effective magnetic sensor to use for the project. It significantly surpasses the hall effect sensors in every way and one thing that is especially important is the fact that the reed switch only needs 5 Gauss time to work compared to the 15 Gauss time that the hall effect sensor needs to work. That fact is important because that means bigger magnets would have to be placed underneath the game pieces if the hall-effect sensors were to be used. The problem with this is that the bigger the magnets are they will possibly attract each other and cause the game pieces to collide and that is something that is not desired for obvious reasons.

Reed switches also significantly make it easier to assemble the project's sensor system because there is no worry and no need to wire them up with each other or to provide a power supply and additional circuitry to it, unlike hall-effect sensors where they do require to be wired to each other, need a power supply, and circuitry to work. Reed switches ultimately saved the group a lot of time and errors to assemble the sensor system.

### 5.1.4 Alternative Sensors

There was research done about other sensors that could've been used and applied to the project, but the biggest factor that we took into account was the cost. One sensor that was found to be vastly superior to the hall effect sensor and even reed switch is the magnasphere. The magnasphere is completely metal and is the most advanced magnetic sensor recently made in the last 70 years. Even though it's the most advanced magnetic sensor and that there is reason to believe that the group should use this sensor, there were many factors that were considered when it came to choosing reed switches or hall-effect sensors over magnaspheres.

**Table 34: Sensors Comparison Table**

| | Hall-Effect Sensors | Reed Switches | Magnaspheres |
|---|---|---|---|
| **Pricing** | – Much cheaper than magnasphere and similar to reed switches in price.<br>– Most are under a $1 each. | – Much cheaper than magnasphere and like hall-effect sensors in price.<br>– Most are under a $1 each. | – Much more expensive than both hall-effect sensors and reed switches.<br>– Cheapest researched was $7 each. |
| **Efficiency** | – Not as efficient and more work to assemble for the project than reed switch and magnasphere. | – More efficient and easier to set up than hall-effect sensors and are perfect for use in the project. | – The most efficient out of the other two magnetic sensors, but it is a bit extreme and not necessary for the project. |
| **Quality** | – More fragile than reed switches and magnasphere.<br>– For the purpose of the project it's ok. | – More fragile than magnasphere, but not hall-effect sensors.<br>– For the purpose of the project it's ok. | – Virtually indestructible, but it is not necessary for the project. |

The comparison of the three sensors within Table 34 indeed does show that the magnasphere is the superior of the three, but it's just a bit extreme, extra, and too expensive for the purpose of the project. They are used for things such as generators and motors and for the project all that is needed is a sensor that can just move around the game pieces and connect with the XY plotter's magnet sensor.

There's no need or reason to pay over $1000 for the 130 sensors that are needed for the project just because they are magnaspheres, when only under $300 can be spent on reed switches or hall-effect sensors which are perfect for the project and are convenient enough. It doesn't matter that magnaspheres are superior in every way because for the purpose of this project the magnetic sensors don't have to be perfect, very advanced, or indestructible. This is just a table-top game that is trying to be built, not anything robotic, industrial, or in the automotive industry where the magnetic sensors need to be very strong, durable, and efficient as possible which is what magnaspheres aim to be.

### 5.1.5 FET Multiplexer/Demultiplexer

In this next section, the different types of FET Multiplexer/Demultiplexers that can be used with the sensors in the project are discussed. When doing research on previous projects like this one like the Cyber Chess project they use a 3 to 8-line decoder, for this project a 3 to 8-line decoder cannot be used. In the Cyber Chess project, they were able to use a 3 to 8-line decoder because a chess board is always in an 8x8 block set up. In Dungeons and Dragons, the board is always in a 16x16 block set up and for this reason a 3 to 8-line decoder can't be used. The decoder that can be used is a 4 to 16-line decoder.

The reason the project needs a decoder is to reduce the number of pins used to information from the reed switches. The decoder will also be used to make it compatible to link the reed switches to the microprocessor that will be used, which will be either the Raspberry Pi or Arduino. The way the 4 to 16-line decoder will work is that it will have the 16 input pins from the reed switches connect to the outputs of the 4 to 16-line decoder and then to the microprocessor.

### 5.1.6 Sensors Summary

As shown and researched above, there is still a debate on whether the group should use reed switches or hall-effect sensors. Ultimately, reed switches are the best one's cost-wise and efficient-wise to use on the project. The alternative sensor the magnasphere that is a vastly superior magnetic sensor compared to both is not necessary for the purpose of the project and is also very expensive. Reed switches and hall-effect sensors end up being a similar price, but the reed switch is easier to set up, doesn't require circuitry or wiring, doesn't need a power supply to work, and are more efficient overall compared to hall-effect sensors.

The largest deciding factor of choosing reed switches over hall-effect sensors is the setup, the efficiency, and the fact that there is more variety and flexibility with reed switches. The easier things are made for the group, the better. This lead to less errors, less complications, and saved time.

## 5.2 LED’s

LED’s were another part of the hardware that was needed for the project. The LED’s were being used mainly for aesthetic purposes to make the board light up and look nice. The LED’s was also used to light up the path that needs be taken and to show where the start and end of the path is for a quest. Also to light up a specific game piece or character when they are the focus of the game at the moment.

The LED’s needed for the project must be programmable so they can be controlled and show the specific pattern or spot that needs to be lit up during the game. They also must be LED strips or LED matrices because it is desired for them to be under each block of the 16x16 board. They also must be cost-effective because there is no need for them to be very advanced or industrial for the purpose of the project. They need to work, look nice, and be individually programmable.

The research done for the LED’s that can serve the purpose of the project showed that individually addressable LED’s would be the best ones to use because they can individually be programmed. There are different types of individually addressable LED’s and they are compared and discussed in this section.

### 5.2.1 Outdoor RGB LED Strip Light Kit

When first researching LED’s this LED strip called the Outdoor RGB LED Strip Light Kit - Weatherproof 12V LED Tape Light was originally being considered because it looked nice and because the group members were not even aware that individually addressable LED strips even existed. The problem with LED strip is that it’s only meant to be used outside for decoration and they are not individually addressable.

This one is only controlled with by a remote and it’s not worth the price either, so this one would not have worked like desired and would be extremely inefficient for the project. With a bit more research the group then found out that programmable and individually addressable LED strips existed and would perfectly serve the purpose of the project.

### 5.2.2 KAPATA Digital RGB Strip

The KAPATA Digital RGB Strip Individually Addressable LED Strip WS2812B WA2812 5m 30IC-30LED/M White PCB Waterproof Dream Color DC5V are one of the LED’s strongly being considered being used in the project because it’s individually addressable, can be programmed, and controlled like we desired. In just one meter of strip 30 LED’s are present and the order comes with five meters of the LED strips. That makes a total of 150 meters worth $33. For the 16x16 board we needed a total of 256 individual LED’s under each square. Just buying two quantities of this product provided the group with more than enough LED’s for the

10x13 board. Since it will bring 300 in total, which is desired because they will provide the group with extra supply if some of the LED's don't work or break in the process. It also can be programmed to flash, strobe, and fade which is also desired for the project. The only problem with these is that they need a bunch of LED drivers as a power supply adding additional costs to the process of setting up the LED's and making them work.

### 5.2.3 Adafruit NeoPixel Digital RGB LED Strip

The Adafruit brand of LED strips are better quality and has less spacing between the LED's than the KAPATA. The Adafruit NeoPixel Digital RGB LED Strip - Black 30 LED performs how it's needed for the project, it's individually addressable, can be programmed, and controlled. It also can be programmed to flash, strobe, and fade which is also desired for the project, just like the KAPATA. The problem with this one is that it's significantly more expensive being $11 for one meter and it has 30 LED's per meter, like the KAPATA. Just three meters of these would cost the same price as the KAPATA that supplies five meters instead.

Only for this reason alone the Adafruit NeoPixel is not worth it and is really not necessary for the project since the KAPATA does the same thing that is needed for the project at a cheaper price and offers more LED's (300), while the Adafruit would only bring 90 LED's for the same price. On top of that, similarly to the KAPATA they also need a bunch of LED drivers as a power supply adding additional costs to setting up the LED's and making them work.

### 5.2.4 Adafruit RGB Matrix for Raspberry Pi - Mini Kit

This LED product: Adafruit RGB Matrix HAT + RTC for Raspberry Pi - Mini Kit was made specifically to connect with the Raspberry Pi which is the microcontroller strongly considered being used in the project, is more advanced and looks nicer than the other two products, and it is also individually addressable as needed in the project. But, it's a very small matrix that would not be useful for the project. It is much smaller than is needed to be for the project, so several would then be necessary and cost wise be very expensive. Only one of these matrices costs $25 with two already being $50 and surpasses the amount of the LED's being considered for use in the project. It really would not be efficient and cost-effective for the project.

### 5.2.5 Flexible 16x16 NeoPixel RGB LED Matrix

After doing more research this might just be the perfect one to use in the project because it is exactly a 16x16 matrix, which is exactly the size of the board for the project. The LED's are also individually addressable as needed also, and because it's a more advanced matrix the patterns can be visually much better and be arranged much nicer and sharper than a normal LED strip is capable of. Significant problems with this LED matrix is the price of being $95 making it much more expensive than all the other LED products that the group has in mind to use and also the fact that even though it is a 16x16 matrix it may not match exactly with the measurements of the board and then be off by a few inches in either the length, width, and height, or all three of them. The only benefit of the 16x16 matrix cost-wise is that it will need fewer LED drivers to power it up, but it doesn't matter since the price of the matrix alone is much more expensive than the price of the LED strips and LED drivers that are needed to power it alone. Therefore, still making the LED Strips the best option to use for the purpose of the project so far.

### 5.2.6 WESIRI 16x16 Pixel 256 Pixels Flexible LED Panel

When doing previous research on LED matrices it was believed that the Flexible 16x16 NeoPixel RGB LED Matrix was the only one of its kind and was not being seriously considered to be used even though it is individually addressable because of it measurements not matching the physical board that is being made and also because of how expensive it is.

The difference with the WESIRI 16x16 Pixel 256 Pixels WS2812B Digital Flexible LED Panel Individually Addressable Full Dream Color DC5V is the price and the matrix are chainable and extendable. It is also compatible with Arduino and Raspberry Pi, which is not decided yet which of these microcontrollers will be used. Having the option to be programmed with either is important. The only problem that was found with this product is that there are no reviews on it in Amazon, but that's just a minor inconvenience that is not important. The group just doesn't want to spend time and money on something that may not work well.

### 5.2.7 CHINLY WS2812b Pixel Matrix, 16x16 256 Pixels

With even more research done on finding 16x16 individually addressable LED's matrices. The CHINLY WS2812b Pixel Matrix, 16x16 256 Pixels WS2812B Digital Flexible LED Panel Programmed Individually Addressable Dream Screen DC5V was found to have the exact same features as the WESIRI LED Panel, except this one has more reviews and the group can put more trust and confidence into this product.

It is few dollars more than the WESIRI brand matrix panel, but it is better to pay a little bit more for a product that is more reliable and that is better quality. The fact that it is chainable, extendable, and can also be programmed by either the Arduino or Raspberry Pi is the part that is most critical for the project. After enough research being done on different LED's it seems like this one will be the easiest to set up and will work just as good as the other ones.

### 5.2.8 BTF-LIGHTING 16x16 256 Pixels LED Matrix Panel

The next and final LED product that was researched was the BTF-LIGHTING WS2812B RGB 5050SMD Individually Addressable Digital 16x16 256 Pixels LED

Matrix Panel Flexible FPCB Dream Full Color Works. The features on this one is no different from the WESIRI and CHINLY, but this one just had more customer reviews than the WESIRI and CHINLY with pictures and even videos to show the quality and performance of the product. Like the CHINLY and WESIRI, it has the most important features needed to work with the project. To make sure that it can be compatible with the board's dimensions and that each individual LED is under each board block it is chainable and extendable.

Also, similarly to the CHINLY and WESIRI it can also be programmed with the Arduino and Raspberry Pi, which is important because like said before the group hasn't decided on which of these microcontrollers to use yet. The BTF-LIGHTING is also priced similarly to the WESIRI and CHINLY, which cost is the other important factor for the group. This being the last product to be researched and being a product commonly used by people shows that the group can rely and put confidence and trust in this product. The reason the group researched brands that aren't popular like these is, so they don't have to spend so much money on a popular brand one, like Adafruit and Neopixel.

### 5.2.9 LED Summary

Out of all the LED's researched one of the cheapest and the one of the ones best suited for the project is turning out to be the BTF-LIGHTING WS2812B RGB LED Matrix. It is one of the best ones in price and does exactly what is desired for the project. It's individually addressable, so it can be programmed as desired to light up the board and make whatever pattern is needed for the game.

It can also be programmed with either the Arduino or Raspberry Pi, which is perfect since the group hasn't decided yet on which microcontroller will be used. Even though the WESIRI 16x16 LED Panel and the CHINLY LED Panel will do the exact same thing as the BTF-LIGHTING LED panel, it has more reviews and more proof that it's a product that will be reliable for the purpose of the project.

Therefore, the group can put more trust and confidence on the BTF-LIGHTING. The Outdoor RGB LED Strip Light Kit is not even individually addressable and costs a lot more than the BTF-LIGHTING. The other matrices, like the Adafruit and Neopixel are not necessary for the project even though they're more advanced and efficient.

The other matrices are too expensive, and the BTF-LIGHTING can do the same thing even if it is not as advanced as the other ones. The best or most advanced and the most efficient LED strips or matrices aren't needed for this project. They just need to work and be individually addressable for the paths, start points, end points, characters, and objects in the game.

| Table 35: LED Product Comparison | | | |
|---|---|---|---|
| | **BTF-LIGHTING Panel** | **LED Strips** | **BTF-LIGHTING Strips** |
| **Pricing** | – The Panel is a similar price to the other LED strips.<br>– More expensive than the BTF-LIGHTING strips. | – All the LED strips researched are more expensive than the BTF Strips.<br>– All around the same price.<br>– Similar pricing to the BTF Panel. | – Cheaper than all the LED strips and BTF Panel.<br>– Only $20.88 for 144 individual LED's. |
| **Programming** | – Each LED can be individually programmed with the Raspberry Pi and Arduino. | – Each LED can be individually programmed with the Raspberry Pi and Arduino. | – Each LED can only be individually programmed by the Arduino.<br>– This is the microcontroller that the group will use. |
| **Customize** | – Extendable only by attaching other LED panels to it.<br>– LED's cannot be cut or extended. | – Can be cut to the desired length.<br>– Each LED can be cut and extended.<br>– This function needed for the board size. | – Can be cut to the desired length.<br>– Each LED can be cut and extended.<br>– This function needed for the board size. |
| **Reviews** | – Many great reviews on Amazon about the product.<br>– Not compatible for the purpose of the project. | – Most LED strips have minimal reviews.<br>– KAPATA even that only had 5 reviews. | – Many great reviews on Amazon about the product.<br>– Exactly what's needed for the purpose of the project. |

Recently, the group bought the LED that was researched to be the best for the project called the BTF-LIGHTING LED panel. After it was received from the mail and physically analyzed it was then concluded that this LED panel cannot be used in the project unfortunately. It was thought that the panel could be cut and adjusted like needed because it was advertised as extendable on Amazon. When physically looking at the panel though it was shown that by extendable what it meant was that it is compatible to connect to other LED panels, not that the panel can literally be extended itself.

Even though a bunch of good alternatives were researched, one of the group members researched an LED strip that turns out will work much better for the purpose of the project. It's called the BTF-LIGHTING WS2812B 144 LEDs/Pixels/m Black PCB Individual Addressable Full Color led Pixel Strip Dream Color Non-Waterproof 3.2FT 1m. In the **Error! Reference source not found.** shown below a comparison between the BTF-LIGHTING LED panel, the other LED strips researched, and the newly found BTF-LIGHTING LED strip is made.

In the table above the comparison of the products is shown and it compares why the LED product that was originally thought to be the best for the project wasn't the best.

The alternatives were considered then after, but after more research on the BTF-LIGHTING brand which is apparently a good brand according to the reviews on their products through Amazon. The BTF-LIGHTING LED strips were found and turns out to be much cheaper than the other LED strips that were researched and the product also having 153 reviews most of them being very good shows that this is a reliable product that can be used for the project.

Yes, it was mentioned that it can only be programmed with an Arduino and the rest of the products can be programmed with both an Arduino and Raspberry Pi, but this doesn't matter. The group originally thought that the Raspberry Pi would be the microcontroller used for this project, but it turns out that the Arduino is the best microcontroller to use for the purpose of the project, so the BTF-LIGHTING LED strip is perfectly fine to use for the project.

## 5.3 LED Drivers

In the previous section, the different types of LED products being considered to use for the project were discussed. In this section, the different types of power supply or LED drivers will be discussed. The LED drivers are needed and are a power supply to turn on the LED's and get them to work. LED drivers can be comparable to transformers for low-voltage bulbs and ballasts for fluorescent lamps.

The reason LED's need to have a specific power supply is because they are made to run on low voltages (12-24V) under direct current, but the problem with most places is that they have higher voltages (120-277V) under alternating current. What LED drivers do is convert the higher voltage that's ran with alternating current to a lower voltage and to direct current. That's why LED drivers are essential as a power supply to LED's.

Another important thing that LED drivers do is to protect the LED's from voltage or current fluctuations. The voltage changing can cause the current to change in the LED that it's supplying. For LED's to work properly they need to stay in a specific current range. The light output in an LED is directly dependent on the current being supplied to it. With all of this explained this means that too little or too much current can cause the light output to be unstable or wear out and raise the LED's temperature.

There are also two different types of LED drivers, they are called internal and external drivers. The one that will be used for the purpose of the project is an external driver. The reason an internal driver is not applicable is because they are used mostly for household lightbulbs. External drivers are the ones made to power the LED strips and matrices that may be used in the project. Now, the different type of LED drivers that can be applied to the project will be discussed.

### 5.3.1 BTF-LIGHTING Power Supply Adapter Transformer

The BTF-LIGHTING DC5V 6A 30W Plastic Power Supply Adapter Transformer for WS2812B APA102 LED8806 WS2801 LED Strip Modules Light is the first power supply that was found because it was suggested through Amazon in the BTF-LIGHTING product page. It is made to work specifically with the BTF-LIGHTING LED Matrix Panel. Sure, it is compatible with the LED product that group is strongly considering using and is also not expensive.

The problem with this product is not any of the features, but the fact that it does not have the best reviews from customers. It also does not come with a power chord. That's not good for the group because they want to make sure that they can get a reliable driver to power the LED Matrix Panel and the group doesn't want to do extra work to get or make a power chord for the driver either, it's not efficient.

That means more research needs to be done to find a better product that can work with the LED's and maybe not just be compatible with the BTF-LIGHTING LED Matrix Panel, but also with the other LED matrices like the CHINLY and WESIRI, in case it doesn't work out with the BTF-LIGHTING LED Matrix Panel.

### 5.3.2 ALITOVE 5V 15A AC to DC Power Supply Adapter

The next product that was found is the ALITOVE 5V 15A AC to DC Power Supply Adapter Transformer Converter Charger 5.5x2.1mm Plug 100V-240V AC Input for WS2811 2801 WS2812B LED Strip Pixel Light. It is compatible with the BTF-LIGHTING LED Matrix Panel and WESIRI LED Matrix Panel since its capacity is 5V and 15A.

It is a bit pricier than the BTF-LIGHTING power supply by ten dollars, but it has a lot more reviews from customers, overall looks like it's better quality, and it comes with a power cord, unlike the BTF-LIGHTING power supply. Therefore, the group can make a better judgment on whether to get this product or not. Still, more research must be done to see if there's an even better product or cheaper product of the same quality that can be used instead.

### 5.3.3 5V 15A 75W Power Supply

The 5V 15A 75W Power Supply 100V-240V or 110V - 220V AC to DC Adapter 5V 15 amp Switching Converter Charger 5.5x2.1mm Plug for WS2811 WS2812B WS2813 2801 LED Strip Pixel Lights only other product found through research that is compatible in every way with the BTF-LIGHTING LED Matrix Panel and WESIRI LED Matrix Panel. Anything beyond 5V and 15A will destroy these LED matrices.

It has the same features and does the exact same thing as the BTF-LIGHTING power supply and ALITOVE power supply, but it has the same problem as BTF-LIGHTING power supply and doesn't have enough reviews from customers for the group to decide whether or not the product will work well or is good enough quality for the project. This and the above LED drivers are only compatible with the WESIRI LED Matrix Panel and the BTF-LIGHTING LED Matrix Panel. There is not much variation between the ones needed for the project because the matrices share the same voltage and current capacity.

### 5.3.4 5V Power Supply, CHINLY 20A

After even more research being done a regulated switching power supply was found from the CHINLY brand call the 5V Power Supply, CHINLY 20A Universal Regulated Switching Power Supply Transformer for WS2811 2801 WS2812B WS2813 APA102 LED Strip Light that can also be used with the WESIRI LED Matrix Panel and the BTF-LIGHTING LED Matrix Panel. With the CHINLY LED Matrix Panel not having a current capacity in the Amazon page different power supplies that are used to power it show and this one was one of them and the one found to be compatible with the other two.

The problem with this product is that the current capacity for the WESIRI LED Matrix Panel and BTF-LIGHTING LED Matrix Panel is 15A. Even though the current can be manually adjusted if it's accidentally adjusted over the 15A it will destroy the LED's, which the group really does not want to happen because they don't want to spend more money to get a whole new set of LED's.

Another problem with this product is or it can be possibly something that works to the project's advantage depending on how the group wants set it up is that it's a universal regulated switching power supply and the way that it's plugged in is different from just a direct to the wall plug-in. Ultimately, it doesn't seem like it's necessary and its bit too advanced for the purpose of the project.

### 5.3.5 LM3402 and Implementation

For visual effects, the LM3402 electronic chip to was considered to deliver constant current to the LED drivers and able to change the brightness level by using a PWM signal from the microcontroller.

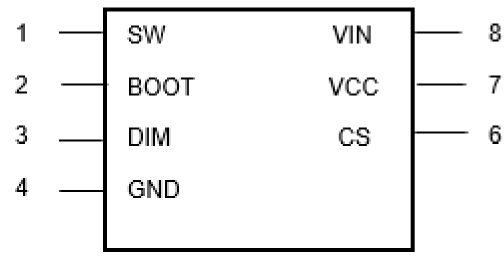


*Figure 17: LM3402 Pin Diagram*

| Table 36: LM3402 Function Table | | |
|---|---|---|
| **Pin No.** | **Function** | **Name** |
| **1** | Connecting to a Schmick Diode and output inductor | SW |
| **2** | MOSFET drive boot trap. Connects to the 10nF capacitor | BOOT |
| **3** | It will reduce the light of LED array using the PWM signal | DIM |
| **4** | Connects to ground | GND |
| **5** | To stabilize light of LEDs | CS |
| **6** | Setting the time light up the LEDs | RON |
| **7** | No need for this pin. Connects to a minimum of 10nF capacitor | VCC |
| **8** | Input voltage (6V to 42V) | VIN |

Once a specific type of LEDs is chosen, the number of LED drivers must be determined in order to determine appropriate number of LM3402's, such as shown in Table 36. Those electronic chips are quite inexpensive so purchasing them is no concern. Once these chips are connected to the LEDs and to the microcontroller, they could change the brightness and colors to add scenery during gameplay.

The Dungeon Master, an individual that can create campaigns and stories for the players, can add brightness and colors depending on the situation that they are experiencing. For example, the players are being ambushed by some enemies; the Dungeon Master adds a reddish or some darkish color to add the mood of the gameplay. If the players find a treasure or some type of reward, the Dungeon Master can add a bright color on the gameboard.

Changing the brightness and adding colors into the game can create a significant impact during the game and getting an interesting experience. However, the group has concluded that this type of electronic chips since the LEDs chosen in the project, the WS2812b, have their own LED drivers and has compatible code in the Arduino library, the NEOPIXEL library.

### 5.3.6 MAX16802B Implementation

For this type of electronic chip, its application is most widely used for industrial lighting was the primary choice for the light of LEDs. However, it can deliver a lot of current which means that it can deliver high-level power into the system. This can cause a safety concern for the participants of the project; therefore this choice can be neglected.

### 5.3.7 LED Drivers Summary

There was not much to be said or researched on any of the different types of LED drivers needed for the project to supply power to the LED's. The WESIRI LED Matrix Panel and BTF-LIGHTING LED Matrix Panel both can use the same power supply because their capacity is 5V and 15A. With the CHINLY LED Matrix Panel it also has a capacity of 5V, but the current capacity is not specified, it's assumed that it's the same or like the CHINLY and BTF-LIGHTING current capacity.

With each of these power supplies only one is needed to power the 130 LED's that will be used in the project. The group made the decision on which of these power supplies to use, they all do the same thing, but one does have better customer reviews than the other ones. In the end they all do the same thing and have the same features, so it's just on the discretion of the group for which one to use for the project.

After the group discovered that the BTF-LIGHTING LED Panel could not be used for the purpose of the project and the BTF-LIGHTING LED strips was used instead this made the decision easier for the group to choose which LED driver to use for the project.

In the link for the Amazon page for the BTF-LIGHTING LED strips a power supply was recommended to go with it called the ALITOVE 5V 10A AC to DC Power Supply Adapter Converter Charger 5.5x2.1mm Plug for WS2811 2801 WS2812B LED Strip Pixel Light. It has 53 reviews on Amazon most of them being positive and is made to be compatible with the BTF-LIGHTING LED strips. There is a Table 37 shown below making a comparison between the LED drivers.

| Table 37: LED Drivers Comparison Table | | |
|---|---|---|
| | **Researched LED Drivers** | **ALITOVE 5V 10A Power Supply** |
| **Compatibility** | – Not all of them are compatible for the purpose of the project.<br>– They all support the voltage that the LED's need.<br>– None of them support the current. | – Most compatible with the LED product that will be used for the project.<br>– This product supports exactly the voltage and current that is needed for the BTF-LIGHTING LED strips. |
| **Usability** | – All the LED drivers were meant for the LED product that was thought would be used in the project. | – The LED driver is meant to be for the current LED product that will be used on the project now. |
| **Pricing** | – For these LED drivers the prices are all very similar. | – The price of this LED driver is similar to the other ones researched. |

As shown in the table above, the decision to use the ALITOVE 5V 10A AC to DC Power Supply Adapter Converter Charger 5.5x2.1mm Plug for WS2811 2801 WS2812B LED Strip Pixel Light turned out to be the best because of the sudden change in LED product that is needed now for the project. Since one BTF-LIGHTING WS2812b 5m 30leds/Pixels/m Flexible Individually Addressable Led Strip Dream Color Non-Waterproof DC5V 16.4ft will be needed for the project because 130 LED's were needed, then that means that one power supply adapter was needed to be able to provide power to all of those LED's.

## 5.4 L293D Stepper Motors and Implementation

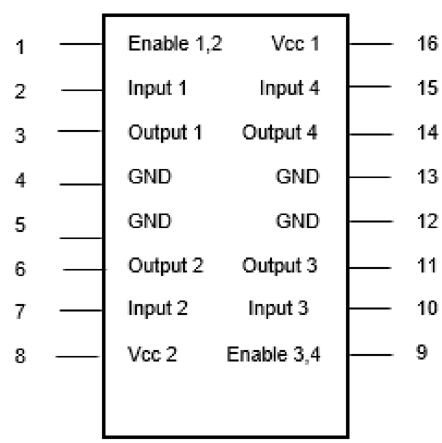


*Figure 18: L293D pinout diagram*

When building the XY plotter, the electrical components were essential in order to move the game piece around the surface. In the beginning phase, the L293D may have been the chip to connect two stepper motors for movement.

**Table** 38**: L293D Function Table**

| Pin No. | Function | Name |
|---|---|---|
| **1** | Enable Pin for Motor 1 (Active high) | Enable 1,2 |
| **2** | Input 1 for Motor 1 | Input 1 |
| **3** | Output for Motor 1 | Output 1 |
| **4** | Ground | Ground |
| **5** | Ground | Ground |
| **6** | Output 2 for Motor 1 | Output 2 |
| **7** | Input 2 for Motor 2 | Input 2 |
| **8** | Supply Voltage for Motors (9-12V) | Vcc 2 |
| **9** | Enable Pin for Motor 2 (Active high) | Enable 3,4 |
| **10** | Input 1 for Motor 1 | Input 3 |
| **11** | Output 1 for Motor 1 | Output 3 |
| **12** | Ground | Ground |
| **13** | Ground | Ground |
| **14** | Output 2 for Motor 1 | Output 4 |
| **15** | Input 2 for Motor 1 | Input 4 |
| **16** | Supply Voltage (5V) | Vcc 1 |

Buying enough L293Ds was feasible to connect the two stepper motors for this project. The reason for this was that when connecting them in multiple ways and regulating power may damage the L293Ds in the long run so it is best to buy them in bulk just in case.

At first, the ULN2003 was the choice for the stepper motors however, the L293D has the advantage of making the motors reverse their positions and they can operate independently due to the H-bridge concept which means that the stepper motors will move clockwise or counterclockwise depending which can add more flexibility in the system when playing the game. By using the PCB software, other things must be considered of what type of a stepper was used for the XY plotter.

## 5.5 ULN2003 Implementation

Previously, choosing the ULN 2003 would've have been an ideal choice for the stepper motors. It was useful when connecting to the Arduino MEGA 2560 due to amplification. It consists eight outputs which each of them has the functionality of amplifying signals from the Arduino MEGA 2560 to large devices. However, it consists of having one sink current which only allows to only work for unipolar stepper motors. In this project, bi-polar stepper motors were considered being used for complex movement.

## 5.6 XY's Plotter Electromagnet

The pencil holder is mostly used in most XY plotter design. However, the pencil holder will change to an electromagnet. The pencil will be replaced with an electromagnet and will be inverted in order to make a near contact with the gameboard underneath. It moved millimeters away from the surface to avoid any magnetic sensor and the touch the surface to the specific game piece and drag the game piece to the designated location by the player by the function of the micro servo. Once it wass done moving the game piece, it will go back to the center of the board millimeters away from the surface.

## 5.7 Microcontrollers

This project executed multiple tasks in one single gameplay, a microcontroller was an important component in building the project. They generally contain multiple pins and each of them have certain function. Since an application is involved which contains a user interface, a microcontroller was connected during this process. Two microcontrollers were considered for this project which are the Raspberry PI and the MEGA Arduino 2560. However, only one of them needed to be chosen for the project. These were possible critical components that connects the user interface and the hardware system which included the stepper motors and motor driver for movement and sensor operations.

### 5.7.1 Raspberry PI and Implementation

Considering codes are going to be used during this project. RASPBERRY PI is an electronic chip that allows for gaming visual effects and moving the XY plotter. It a low-cost microcontroller and contains a decent memory.

This can be widely used on the visual effects and media which are essential into the game. It might as well include HDMI quality when illustrating on the users' gaming application when showing the HP and Armor Level with a custom interface. Also, this microcontroller can manage sound effects and distinctive audio in the game. It can also selectively light up LEDs and change the brightness with the help of LM3402 electronic chip. It can send commands to the L293D to activate the Stepper motors in order to move the XY plotter.

At first, the other microcontroller was MEGA 2560 due to multiple features, however, Raspberry PI microcontroller is familiarized by computer engineers in this project. They have a strong foundation of Raspberry PI's programming code and Raspberry PI can behave like a little computer and it can manage high level power which is necessary for this project.

For this microcontroller, the appropriate coding framework was phyton rather than using C programming. The reason for this is that the computer engineering students are familiar with the modules and object instructions for this type of code. That being mentioned, the Raspberry PI is the most efficient microcontroller for this project.

### 5.7.2 Arduino MEGA 2560 and Implementation

The Arduino MEGA 2560 was another choice as a microcontroller. It does not have as many external pins as the Raspberry PI, but it can function the same way and accomplishing the specifications of the project. When there was an external power connected to the Arduino MEGA 2560, the board can operate up from 6V to 20 V, but the recommended to avoid overheat and damage to the board is 6V to 12V which is more than enough for the project to work especially when the stepper motors were involved.

The problem would arise when connecting to the LEDs, motor driver, and the Stepper motors. By using the PCB software, a schematic was developed with the Arduino MEGA 2560 along the connections of these main components. Also, it has a feature that can protect the USB from damage.

There was no significant modification on the board when it came to connect the LEDs, however, the board was dealing with the stepper motors which meant producing higher current. From that, an IRF520 MOSFET needed to be implemented with a 1N4007 diode connection. This MOSFET controlled the bigger loads when initiating the stepper motors on the XY plotter and the diode will be used to protect the overall circuit. This was very effective on using the PWM signal for these types of operations and this Arduino MEGA 2560 provided this process.

### 5.7.3 Raspberry PI Vs. Arduino MEGA 2560

The Raspberry has a lot of functional pins that were not going to be used in this project due to limited time. Raspberry Pi was described to be a “mini-computer” due to its complexity. After having advices from experts and online articles that elaborate more on the Raspberry PI, it was concluded that this microcontroller is not a viable option for this project even though the requirements are achieved by it.

The problem of this situation was that the microcontroller can do much more rather than doing simple functions for a higher price which can be counterintuitive. Therefore, the MEGA 2560 was an Arduino microcontroller that fits for this project. This type of the microcontroller is meant to do simple functions that were necessary for the project. The main goal was to achieve the requirements and implement them as simple as possible in order to budget and avoid complexity during the making of the project.

AT MEGA 2560 consumed significantly less than the Raspberry PI which means there was no problem in setting high voltage in the project. Also, Raspberry PI can do a very well job in using for HD screens or large screens since it contains a HDMI, however, this project consisted of no such thing. For the durability, the AT MEGA 2560 was the right choice for the project making.

## 5.8 XY Plotter's Stepper Motors

These XY plotters are widely used for printers since the main functionality is drawing or scan documents. However, the XY plotter would be used just to have an electromagnet which was pointed upwards the base of the transparent surface.

The XY plotter moved the game pieces just underneath the 10x13 opaque surface. The essential components are the step motors for smooth movements on the gaming board. Since the stepper motors simultaneously moved the game piece to the designated location rather than moving it in a rigid motion and they are extremely reliable.

Two stepper motors were assigned to move in X and Y motion respectively and they were moved in different rates and deliver the game piece to the designated location once the electromagnet grabs the game piece. DC would have the choice, however, once assigned to their respective axis of motion, they caused the game piece to move rigidly. For the stepper motors, that was not the case since they can set different rates of speed which can cause the game piece to move smoothly.

The next motor would have been the servo motor, this motor could have been the ideal choice for this project due to precision in terms of acceleration and velocity, however, the complex circuity for this device is quite overwhelming and requires a difficult search to acquire such a device so the most feasible choice is still the stepper motors.

Also, the stepper motors have external pins that can be used for the microcontroller which can essential for the project since the connection between the user's application and the XY plotter was used by using the AT MEGA 2560 microcontroller.

Another significant consideration is the stepper motors are needed to be unipolar since they are easy to configure, there is no need for a bipolar stepper motors since they are complex to connect and difficult to drive them. With the help of the L293D, it consists of a H-bridge circuit which can help the unipolar stepper motor to have better torque and efficient.

### 5.8.1 Appropriate Frame Size for Stepper Motors

The XY plotter required amount of torque and adequate speed for the game pieces to move around and other components. The bigger the frame size of the stepper motor, the higher amount of torque, speed, and power it can deliver. In this case, the XY plotter was required only a medium-sized stepper motor to accomplish medium speed and medium-level power. The stepper motors were measured in terms of NEMA. The appropriate stepper motor that can be used for this project was NEMA-17 in terms of price and the amount of ‘holding torque’ it provided in this project, with this shown below in Table 39.

| **Table 39: NEMA 17 Specifications** | |
|---|---|
| **Holding Torque** | 59 N-cm |
| **Rated Current Per Phase** | 2 A |
| **Frame Size** | 42x42 mm |
| **Rated Voltage** | 12-24 V |
| **Resistance per Phase** | 1.4 Ohms |

This was the optimal choice because it was within the project budget and the frame size was adequate for the XY plotter assembly. Another thing to consider was the power supply per stepper motors. The rated voltage was the important parameter to choose a chopper driver controller.

### 5.8.2 Stepper Motors’ Power Supply/Chopper Driver Controller

The power supply for the stepper motors was critical for operation. The power supply voltage had to be slightly higher than the rated voltage of the stepper motors for better performance in terms of speed and torque. However, the more voltage and current the power supply delivered, the more heat will dissipate on the system which it causes to overheat and damage the system. It was best to choose a stepper motor with adequate voltage limit in order to avoid this possible predicament. They cannot connect straight to the power supply; the chopper driver controller must be present to avoid damage to the stepper motor.

The chopper driver controller can give high constant current to give the stepper motors more torque and keep stepper motors safe. They can ‘chop’ unnecessary high voltage that can damage the stepper motor.

It was imperative to choose carefully what type of chopper driver controller was necessary to use a specific stepper motor. In order to avoid any damage, important specifications of the stepper motor were considered such as “amps per phase” – how much current can the stepper motor can withstand without overheating and “resistance per phase”-resistance per phase. Calculating these two parameters gives voltage. Usually these chopper driver controllers are rated by voltage.

### 5.8.3 TB6600 Stepper Motor Driver

By general knowledge, stepper motors cannot be connected straight from the power source. Between the power source and the stepper motors, there was a chopper driver controller so that the stepper motors can operate. The various specifications are shown in Table 40. This chosen stepper motor driver was bigger than the stepper motor themselves, however, this was not vital for the project because this motor driver and accomplish well with the NEMA 17 stepper motor.

| Table 40: TB6600 Stepper Motor Driver Specifications | |
|---|---|
| **Current Control** | 0.5-3.5 A |
| **Frame Size** | 96x56x33 mm |
| **Rated Voltage** | 9-42V |

This is generally used by most stepper motors due to specifications in terms of current and voltage. The location of this stepper motor driver was a challenge in this project. The best way to accomplish this was to put the motor driver next to the stepper motor. Based on this model, two motor drivers were required for the two stepper motors.

Once the XY plotter was assembled, the motor driver and the stepper motor had to be right next to each other and  was placed on the edges of the XY plotter to avoid physical interruption when the gameplay takes place. Another thing to consider was to be aware how the power source was connected to the motor drivers. The connection between the power source and the motor drivers was modified.

### 5.8.4 Motor Controller

The motor driver could’ve been the ideal choice for the project due to cheap pricing. However, the disadvantage is to put so much thought in building the circuit with the motor driver. The motor driver did not have feedback like the motor controller if an error occurs.

Even though the price for a motor controller was relatively higher than the motor driver, the motor controller was the choice for the project due to feedback from the functions of the stepper motors when moving the game piece on the 10 X 13 grid.

The feedback system allowed for the motor driver to correct the devices themselves when an error had occurred which was an effective use for the project.

### 5.8.5 Pololu DRV8833 Dual Motor Controller

The right choice for the project had a motor controller due to its feedback functionality. It corrected itself if the execution is incorrect due to the interference with other magnetic sensors or the code in general when testing the XY plotter. The Pololu DRV8833 was chosen to execute physical movements from the XY plotter by using NEMA 17 stepper motors. This specific type of this motor controller is chosen when following the parameters. The rated voltage of the motor controller must be higher than the rated voltage of the stepper motors. This dual motor controller operated only one bipolar stepper motor, therefore, there were two motor controllers, which is shown .in Table 41

| **Table 41: Pololu DRV8833 Dual Motor Specifications** | |
|---|---|
| **Current Control** | 13 A |
| **Frame Size** | 75x43 mm |
| **Rated Voltage** | 5-30V |

## 5.9 A4988 Stepper Drivers

The stepper motors on the assembled XY plotter required stepper drivers in order to modify a fixed movement or the step motion during gameplay. For these stepper drivers, the chip had to be configured in full step motion by having an active low on MS1-3 pins on the schematic.

**Table 42: A4988 Stepper Drivers MS's pin Configuration**

| **MS1** | **MS2** | **MS3** | **Microstep Resolution** |
|---|---|---|---|
| Low | Low | Low | Full Step |
| High | Low | Low | Half Step |
| Low | High | Low | Quarter Step |
| High | High | Low | Eighth Step |
| High | High | High | Sixteenth Step |

From these choices mentioned above, the A4988 stepper drivers were essential components for the project, not because they have a functionality of configuring microstep resolution, but they have a flexible potentiometer to begin with. The potentiometer was able to configure for appropriate current to pass through to the stepper motors in order to operate efficiently.

## 5.10 3-5V Bi-directional Logic Level Converter

The ESP8266-12E is a wifi module that was responsible for online communication between the user and the board of the XY plotter. Since the operating voltage for the Arduino Mega 2560 is 5V and the module required 3V, a bi-directional logic level converter was needed. This component was required because the module needed to receive and sent data to the Arduino Mega 2560.

The previous method was implementing a voltage divider for cost-effective purposes, however, sending data from the ESP8266/12E to Arduino Mega 2560 during normal operations in 3V to 5V was not possible. The HV pins were connected to Arduino Mega 2560 respective pins along with a 5V pin. For LV pins, the ESP8266/12E were connected to its respective pins with a 3V pin.

## 5.11 Configuration

In the project, the configuration was initiated by connecting the appropriate chips so they can do individual tasks in the XY Plotter as well the assembly. The AT MEGA 2560 microcontroller was connected to the LED drivers and the L293D electronic chip in order to move the stepper motors. LEDs were placed right underneath the surface connected to the microcontroller AT MEGA 2560.

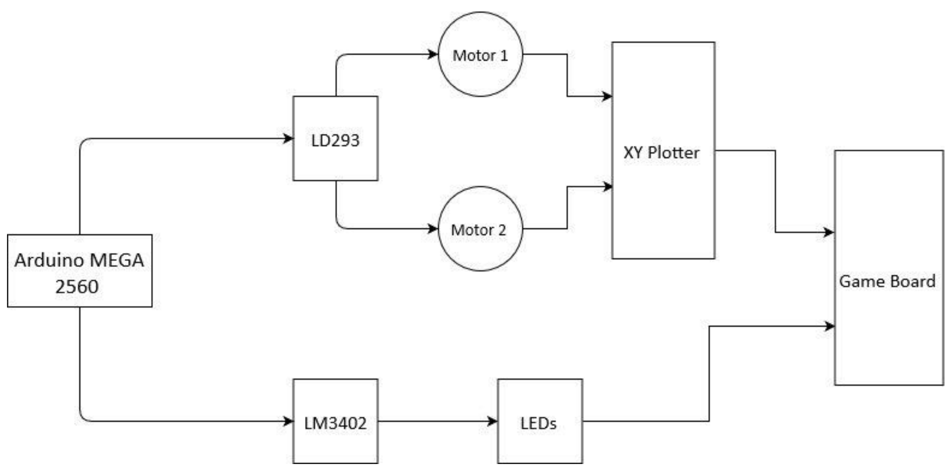


*Figure 19: Simple hardware configuration*

The XY plotter was right underneath the opaque surface (game board). The AT MEGA 2560 was considered the main microcontroller and connected the LD293 and the LM3402 electronic chip which they can do their tasks respectively. The AT MEGA 2560 received all data commands from the players and the Dungeon Master and sending those commands to its respective electronic chips. The LD293 chip were used for movement and the LM3402 for visual effects. This whole system was operated by using the linear power supply. The Figure 19 represents the general flow of the configuration.

Most components were connected to the AT MEGA 2560 microcontroller from which, it sends commands from. The great advantage about the AT MEGA 2560 is that it consists multiple connections and ports and have a function of a small version of a computer.

Since the project consisted of multiple features including audio interface, color brightness for each multiple scenarios of gameplay, and directing command movements to the stepper motors which then move the electromagnet that can hold or attach the game piece and drag from one position to another at the time of the player's choosing during his or her turn.

## 5.12 Testing

For the hardware aspect of the project, it was essential to test the connectivity between AT MEGA 2560 and the electronic chips for optimal performance. The XY plotter was tested by selecting a location on the gameboard so that it can carry that game piece to that location. The connectivity between the stepper motors and the data chip LD293 was then altered if the XY plotter did not have the capability to transport the game piece to the designated location. The LEDs were tested by the Dungeon Master to use a specific color to see if the LM3402 can receive that command from the AT MEGA 2560.

The modification for the XY plotter was having an electromagnet and getting close to the gameboard as close as possible. The electromagnet must be intact when the XY plotter starts moving the game piece and observe if it can operate appropriately when selecting specifically the game pieces whilst avoiding other magnetic sensors therefore, the connection between the AT MEGA 2560 and the electromagnet must be tested for the project to work.

Since the project was required to use a high-level power, it was imperative to take extreme caution when testing the linear power supply in order to power the XY plotter and the LEDs. To test this aspect, a small prototype of the project was to created which consists of small number of LEDs, LED drivers, along with a spare Arduino MEGA 2560 and ran a sample gameplay scenario to observe their functionality.

# 6.0 Project Design Summary

Whenever the XY plotter was to be purchased or to be assembled, some steps were needed to be taken account in regards of configuring the XY plotter which included the stepper motors, motor driver, and the use of the AT MEGA 2560. In the software section, the interface was going to be developed for gaming players and the development of the data path that took place on the 10x13 grid for the movement of the game figures during gameplay.

## 6.1 Hardware

The connection of the magnetic sensors, LEDs, and the stepper motors were vital for the project. The assembly of XY plotter consisted of two stepper motors, an electromagnet, and other parts which then were dependently connected to the electronic chip LD293 to generate movement of the XY plotter with the help of the stepper motors which ultimately were connected to the microcontroller.

For the LEDs, the AT MEGA 2560 connected the LM3402 which is the appropriate device to connect to the LED drivers and then to the LEDs. The LEDs are located underneath the gameboard since the surface itself is opaque for visual effects. For the power source, the AC-to-DC linear power supply involved a transformer for the change voltage level, rectifier, and numerous capacitors along with a voltage regulator.

## 6.2 Software

In the software aspect of the project, coding took part for the AT MEGA 2560 to function including place movement by using data path which created multiple algorithms or possibilities of movement and choosing the optimal path for the game piece to move around the obstacles that can be created on the 10x13 grid game board.

When the player can choose a location to move, the AT MEGA 2560 microcontroller sent programming algorithms for sensing communication which then communicated to the XY plotter to choose a game piece and move to a certain location. Also, the microcontroller displayed a visible interface indicating health and armor level of the character created. From there, the interface stored and modified data to upload certain statistics after finishing a battle or doing some certain action during gameplay.

# 7.0 Project Prototype Construction

If assembling the XY plotter was the right choice for the project, then the main components were needed to be taken account when building it. Parts were going to be acquired when assembling the XY plotter and is shown in the following tables and a summarized procedure on how to assemble the XY plotter.

## 7.1 Assembly

Building the XY plotter required multiple components including the stepper motors, motor controller, the electromagnets, LEDs, and the Reed switches. This was a choice in order to avoid any waiting time after purchasing an XY plotter. Multiple main steps were considered in order to assemble the XY plotter. This assembly process is described in the **Error! Reference source not found.**.

The placement for these components would be a challenge. The AT MEGA 2560 and other main components would be placed on the edges of the XY plotter and building some type of cover walls to hide those components for the project to be more presentable which is referenced in figure 3. In the figure 3, it shows the whole outline of the project.

Ideally, the Reed switches on were placed on the top of the XY plotter with a transparent surface of 10x13 grid gameboard. Once this project was built for presentation, the weight was an overwhelming concern since so many components were involved and the extra weight from the XY plotter added to the overall weight and dimension, therefore extra caution is needed.

| Table 43: Assembly Steps for the XY Plotter and Additional Components | |
|---|---|
| **1** | Acquired parts and main components |
| **2** | Started building the outer layout of the XY plotter |
| **3** | Confirmed that the XY plotter should be nearly equal to the 16x16 grid |
| **4** | Converted the pencil holder to a magnetic holder |
| **5** | Applied a stepper motor (NEMA 17) on the side of the XY plotter |
| **6** | Applied another stepper motor (NEMA 17) on the opposite side of the XY plotter |
| **7** | Connected the each L293D electronic chip to each stepper motor |
| **8** | Connected two chopper driver controller (TB6600) to each stepper motor respectively |
| **9** | Acquired LEDs and Reed Switches for the 16x16 grid |
| **10** | Connected respective LED drivers to the LEDs and place a Reed switch for each individual square grid |
| **11** | Connected for each LEDs for Pulse Width Modulation (PWM) |
| **12** | Connected additional wires from the LED drivers |

## 7.2 XY Plotter Assembly Vs. Purchasing

The assembly of the XY plotter was beneficial to have a strong comprehension on how the XY plotter works, however, the time to assemble and to test its features was going to take longer than expected even if outside help was involved. For that reason, purchasing XY plotter was the only viable option due to limited time given for this project. The ideal parameters of the XY plotter had to be slightly larger or equivalent to a 10x13 grid. It was imperative to see if the XY plotter can be modified to satisfy the conditions and requirements for the project. The types of XY plotter were also a factor to consider as well.

### 7.2.1 EleksDraw XY Plotter Pen Writing Machine

Since buying the XY plotter was the most appropriate option in doing the project, the EleksDraw XY plotter was considered being bought and used for testing when connecting the AT MEGA 2560. However, the size was quite smaller than expected. The working size of 11 X 7.9 and the cost were quite high if the purpose of buying the XY plotter was to test the pin functions from the Arduino. This product needed to be modified including changing the stepper motors to NEMA 17 stepper motors and the magnetic holder as well.

### 7.2.2 ETE ETmate XY Plotter

For this XY plotter, the ETE ETmate is slightly bigger than the previous product. The working area for this specific XY plotter is roughly 8.3 X 11.7. However, this product is more accurate which was essential to the project when transporting the game pieces on the gameboard and a linear power system was not required. With that considered, the power source would be 12V DC. C programming language is needed for this type of XY plotter.

### 7.2.3 Autek Laser Engraving Machine

After searching three products that were nearly satisfying technical parameters which were voltage requirements and the grid size. Since the XY plotter had to be bigger than 10 X 13 to work, the Autek Laser Engraving Machine XY plotter was considered the right choice for this project. The working size of this XY plotter is the 17.7 X 17.7 which is bigger than 10 X 13 to work with. The problems would've arise when configuring this XY plotter to have a magnet holder and a micro-servo installed.

### 7.2.4 IDRAW Machine XY Plotter

This XY plotter required 12V and 1A for it operate efficiently. It has a lighter weight compared to other XY plotters and stepper motors are easily assembled. Even though cost was higher than the other XY plotters, it achieved the ideal working area that the project required. Therefore, this XY plotter was the choice for hardware configuration.

## 7.3 Project's Transportation/Storing

Once the project was built and tested, the temperature was considered when storing the project in a house. The temperature in Florida is usually very hot which can damage the circuits all together. The way this was prevented from happening, the ambient temperature was calibrated to 65 degrees Celsius for storage. The project was kept away from pets and moist environments for safety purposes. The location to where the project was at a team member's house due to space. Since the project was built by 10X13 size, the room provided for Senior Design was small and inconvenient for the project. The size of the doorway was too small to get the project out of the room once it was built and completed. Therefore, the place was in one of the team member's location.

Due to the construction of the project, the weight was too overwhelming to transport for presentation. In order to easily transport the project, a platform constructed slightly bigger than 10x13 size can be built along with four wheels on each side, but it wasn't necessary. Since the project was constructed in one of houses of the participants, a car was necessary to deliver it to the presentation. To lift the project to the car, it simply required all four participants to do such action. These parts acquired were subject to change in order to minimize the weight of the project to avoid any complications regarding transportation.

## 7.4 Power System AC to DC Converter

The game board will come with many electronic components that will require a DC voltage supply. The power supply purchased may not have enough watts to power up all components hence an AC to DC converter circuit will need to be researched to guarantee all parts have enough power to operate.

### 7.4.1 Power Cord

To start off, the first thing needed is an AC power cord. It should come with a standard plug that will fit into any wall outlet. The power cord should be long enough (estimate 6-9 meters).

### 7.4.2 Power Transformer

The transformer was needed in the converter circuit design since the input voltage from any regular wall outlet is 120v and this may be higher than the voltage required for any of the components such as the microcontroller and electromagnet on the game board, therefore a transformer may be needed to bring the voltage down to a more desirable and manageable level for the electrical components.

### 7.4.3 Rectifier

A full wave rectifier was needed for this circuit, it consists of two diodes with a transformer or a diode bridge made up of four power diodes which is called a bridge rectifier. For converting from AC to DC signal, a bridge rectifier has more advantages than a full wave rectifier since it does not require a special centre tapped transformer which in return reduces the size and cost for their project. The single secondary winding is connected to one side of the diode bridge network and the load to the other side.

### 7.4.4 Voltage Regulator

A voltage regulator generated a fixed output voltage of a preset magnitude that remained constant regardless of changes to its input voltage or load conditions. Even though after proceeding with all the previous steps most of the signal was already a steady DC signal, the many variations in voltages can potentially was to harm the components in their board that were being powered up hence why a voltage regulator was necessary. It consisted of a resistor in series with a diode, the purpose of the resistor was to dispose of any unnecessary voltage from both the capacitor and rectifier. The resistor and diode also took the signal input from the rectifier and smooth the signal out into their desired voltage output.

After the group had successfully set up the transformer, rectifier and voltage regulator. The expected voltage now met the voltage required for the components in their gameboard to work such as the microcontroller, LED, sensors. If needed, two AC to DC circuits were used if the motherboard power supply was not enough to power the components mentioned below.

## 7.5 Alternative Power Source 12 V DC

The AC-to-DC converter was the viable option for the project, however, the assembly for this circuit created potential electrical risks and safety. This task would be somewhat a distraction for the real focus and objective for the project.

Most of the focus was on configuring on a XY plotter and implementing a data base which included web server connection on a 10X13 grid. The LED drivers have their own power source and each of the them can power up to 150 LEDs which was more than enough. Also, the cost would be relatively expensive especially when constructing a transformer component in the power system.

# 8.0 Project Prototype Testing

The idea of testing this project was to produce a small scale XY plotter that tested its functions along with the connection with the AT MEGA 2560, the LEDs, and the stepper motors. A few scenarios of gameplay took place and tested the movements of the game figures on a smaller size than 10x13 grid.

## 8.1 Hardware Test Setup

The procedure was to test the LEDs and the LED drivers while using the AT MEGA 2560 once the code and the software setup had taken place. This was to see how the microcontroller worked when a user, the Dungeon Master in this case, can manipulate the level of brightness and the intensity of the color from the LEDs before placing the LEDs on the transparent 10 x 13 game board.

When the XY plotter was assembled and the L293D was connected to the stepper motors. The AT MEGA2560 was connected to it and it ran sample run of code to see if the XY plotter dragged the game piece to the specific coordinate. The power supply was carefully observed since it may have drawn too much heat into the system when operating the XY plotter.

### 8.1.1 Sensor Setup

The sensors being used for this project called reed switches were setup as a 10x13 matrix, which means that 130 reed switches were needed to make the matrix. After research was made it was discovered that to set up the matrix using reed switches that it also needed diodes. The specific type of diode that was used in the reed switch matrix is called 1N4148. Other components that were also needed to make the reed switch matrix were as follows in Table 44.

**Table 44: Reed Switch Component Table**

| | Quantity | Component |
|---|---|---|
| **Reed Switches** | 130 | Reed Switches |
| **Diodes** | 130 | 1N148 Diodes |
| **Resistors** | 10 | 10 kOhm resistors |
| **Shift Register** | 1 | 74HC595 |
| **Shift Register** | 1 | 74HC165 |
| **Microcontroller** | 1 | AT Mega 2560 |

As shown above, all these components were needed to make the reed switch matrix work, it was specifically being used to be programmed with the AT Mega 2560. The reason why the reed switches needed diodes was because “ghost” connections can occur as in the microcontroller mistakes some of the reed switches as being activated when they’re not. The diode is what takes care of the problem by it being placed in series with the reed switches.

The resistors were needed to be able to connect the reed switches-diodes to the Mega 2560. As a way of making it possible to create the reed switch matrix one of the group members created a PCB that made it possible to connect the components to the AT Mega 2560. This was all tested on a small scale before it was built on a large scale to make sure that it worked and was compatible for the purpose of the project. Below is a diagram on how the reed matrix by itself looks like.

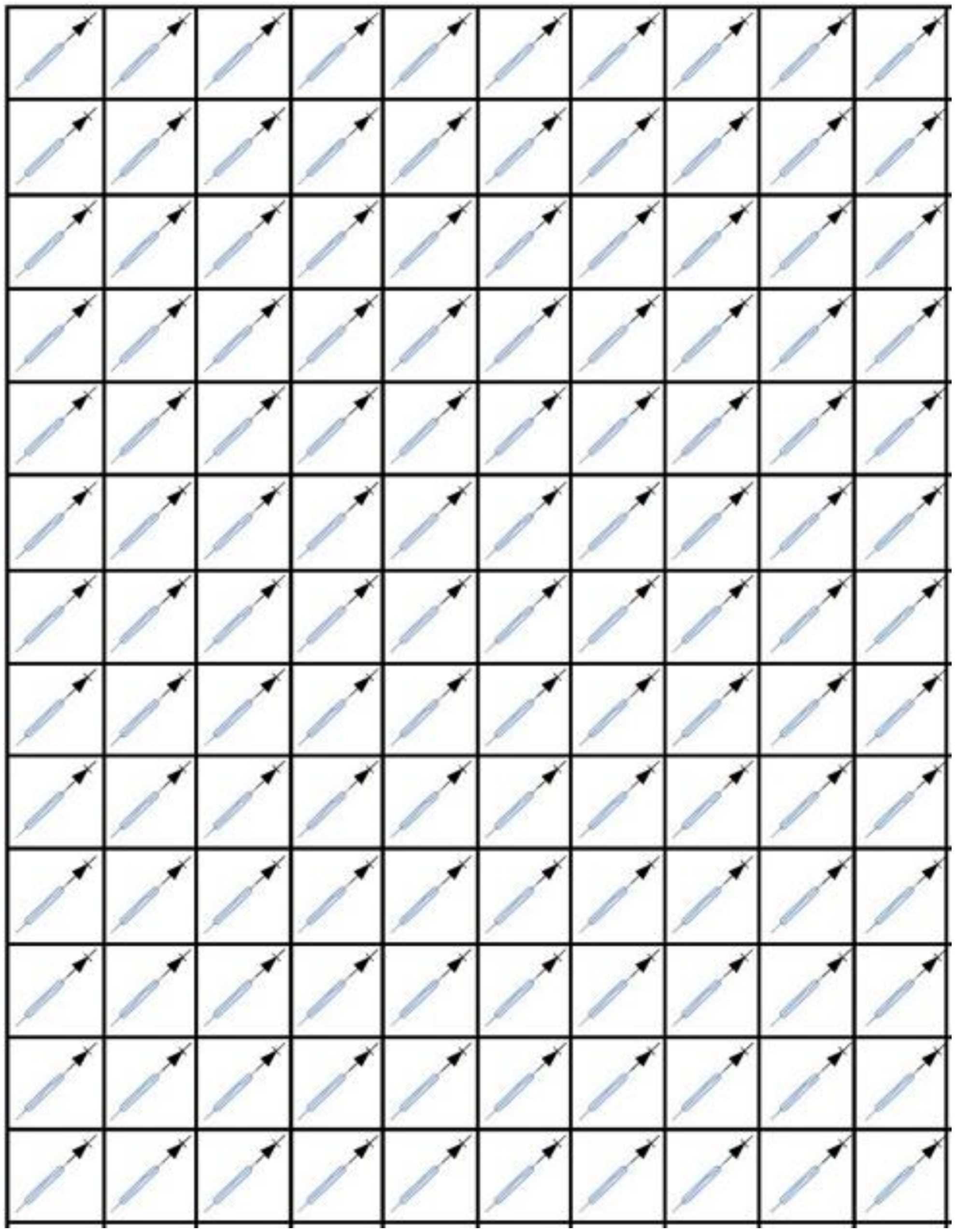

*Figure 20: Reed Switch Matrix*

As shown above within Figure 20, this was exactly how the reed switch matrix looked like once it was assembled. As noticed, there were a lot of reed switches and diodes in this matrix. It took the electrical engineer students a lot of time to be able to assemble this matrix and they even needed help from the computer engineers in the group to also help assemble it.

Connecting it to the resistors and then the AT Mega 2560 did not take too long in comparison to how long it took to assemble the matrix. This was the most efficient way to have the matrix set up and the reed switch matrix was still be easier to set up in comparison to the hall-effect sensor which was the alternative that was discussed in the next paragraph.

An alternative to the reed switch matrix if it hadn't turned out to work well for the project would've been to then make a hall-effect sensor matrix instead. It would've taken more circuitry and wiring to make it work, but it was a very good alternative in case the reed switch matrix didn't work or turned out to be more complicated than the group thought. As the comparison that was made before shows that hall-effect sensors are not as modern and efficient as reed switches, but in theory they were perfectly fine to work for the purpose of the project. Below is Table 45, which shows what components would've been needed to make the hall effect matrix work.

**Table 45: Hall-Effect Sensor Component Table**

| | Quantity | Component |
|---|---|---|
| **Sensors** | 130 | Hall-Effect Sensors |
| **Resistors** | 23 | 10 kOhm resistors |
| **Shift Registers** | 2 | Shift Registers |
| **Microcontroller** | 1 | AT Mega 2560 |

As shown above, there were less components needed to make a hall-effect sensor matrix, but there were more things needed to assemble it and put it together. For example, all the hall-effect sensors need additional wiring up to each other to make the matrix and they need additional circuitry, and an active voltage and current on always for it to work. In the end, the group only considered hall-effect sensors only if it turns out that the reed switch matrix was more complicated and costly to setup, which it was not.

From the research found and assembly done, the reed switch matrix turned out to be easier to set up and less expensive to set up. The hall-effect sensor was strongly not recommended for the group to build because of how long it would take to build and because of all the wiring and circuitry that it would need. Below a diagram is shown on how the hall-effect sensor matrix would've been set up in case it was needed, and it more clearly shows why it was not recommended and how it would've been harder to set up than the reed switch matrix.

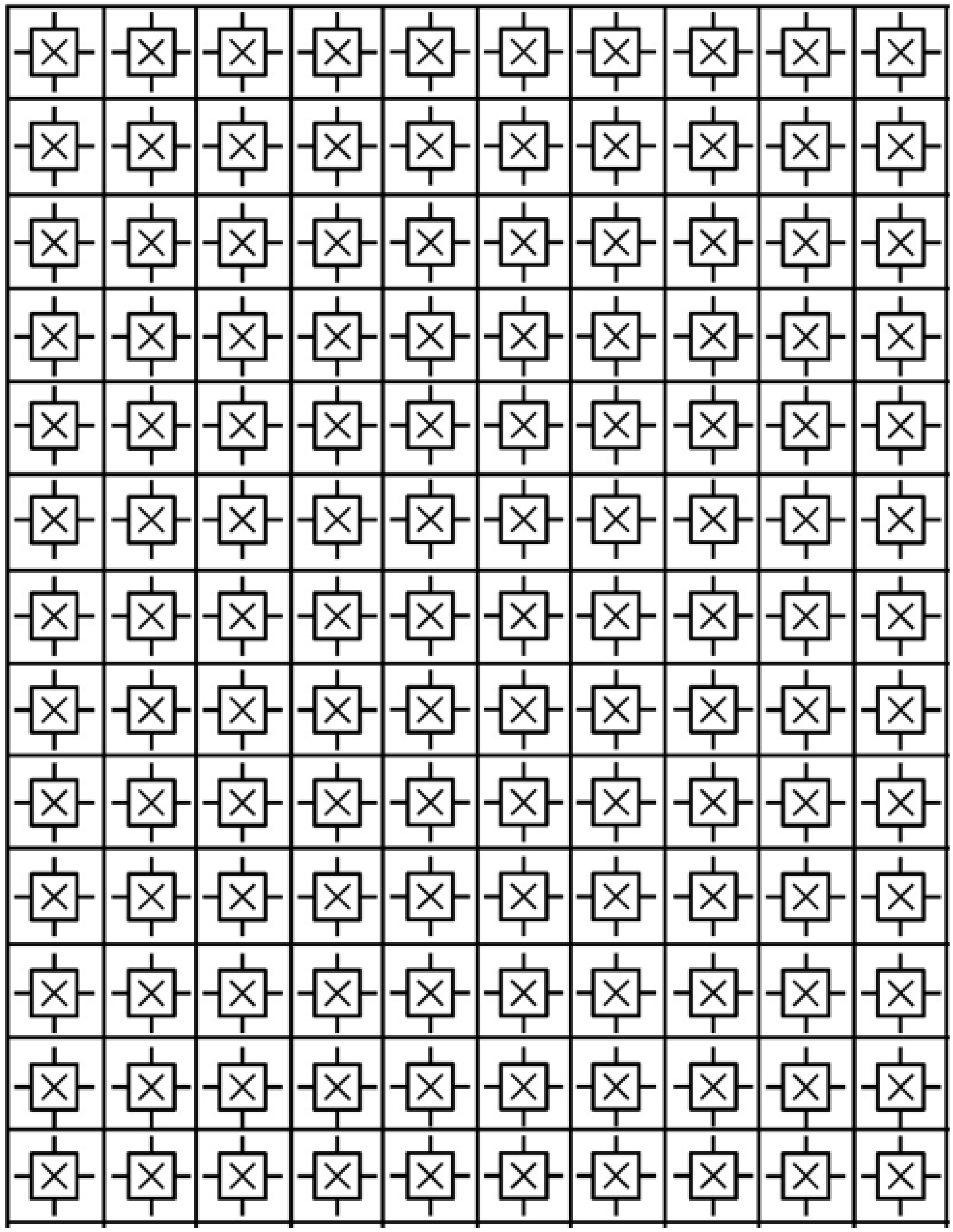

*Figure 21: Hall-Effect Sensor Matrix*

Shown above within Figure 21 was exactly how the Hall-Effect Sensor Matrix would look like if it was assembled. The only thing that was not shown in this matrix was all the wiring that would be needed to connect the hall-effect sensors in sequence to each other. The hall-effect sensor matrix would need more resistors to be able to connect to the power supply circuit, it would also need more than just one power supply to connect all the resistors needed to power all the hall-effect sensors. In previous projects, like this one it was not much of a hassle to use hall-effect sensors because these projects have only used and needed an 8x8 matrix.

That while still challenging is not too difficult or time consuming to set up because only 64 hall-effect sensors were needed to set it up. For this project 130 hall-effect sensors would be needed. It took a long time to set up the reed switch matrix which doesn't even require wiring or a power supply to work. It's obvious that a hall-effect sensor matrix was to be avoided, which it was.

### 8.1.2 LED Setup

The way the LED's were set up are by cutting the LED strips to the length as needed to match the dimensions of the board. Each individual LED was placed under each individual square.

The LED's then were set up in a matrix and then they were connected to LED drivers that supply power to them, then the LED's were also connected to the microcontroller to be able to program each individual LED. In general, there was not much needed to set up and make the LED's work.

The challenge came with programming it with the microcontroller and being able to connect all the LED's to the ports available in the AT Mega 2560. Table 46, which is shown below, lists everything that was needed to set up the LED's.

**Table 46: LED Component Table**

| | Quantity | Component |
|---|---|---|
| **LED Product** | 1 | BTF-LIGHTING LED Strip |
| **Power Supply** | 1 | ALITOVE 5V 10A Power Supply |
| **Microcontroller** | 1 | AT Mega 2560 |

As shown on the table above there was not much needed to set up the LED matrix for the board. One of the BTF-LIGHTING WS2812b 5m 30leds/Pixels/m Flexible Individually Addressable Led Strip Dream Color Non-Waterproof DC5V 16.4ft was more than enough for the matrix since only 130 were needed to go under each square on the board and it's great that it brought 20 extra LED's in case it was necessary to have more.

It was important that the LED's went under each square on the board to illuminate the whole board. The purpose was for them to look nice and light up pathways that were taken in the game.

One ALITOVE 5V 10A AC to DC Power Supply Adapter Converter Charger 5.5x2.1mm Plug for WS2811 2801 WS2812B LED Strip Pixel Light was also more than enough for the project because the LED strips can only handle up to 5V, any more than that will burn out the LED strips. Then, to program each individual LED it will be connected to the PCB. The LED's made the board really light up and look nice.

The LED's aren't really needed, but it was just for aesthetic purposes. They were also programmed to light up the path of the game and where the characters were going. Below is a diagram showing how the LED matrix was set up and looked like.

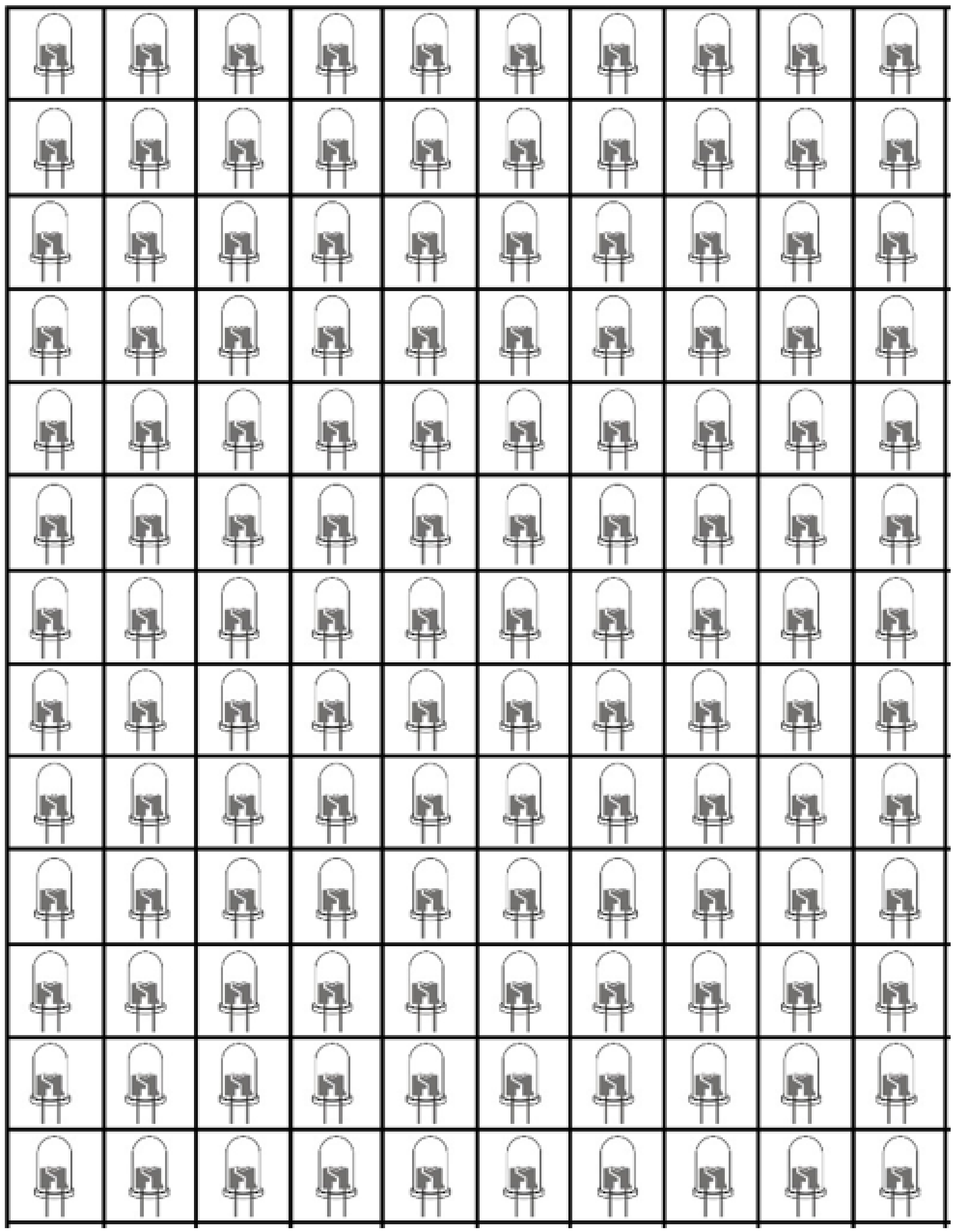

*Figure 22: LED Matrix*

As shown above within Figure 22, the LED matrix was very straightforward. There was not too much to it because all that is needed to assemble the matrix was the correct length cut since it's the BTF-LIGHTING WS2812b 5m 30leds/Pixels/m Flexible Individually Addressable Led Strip Dream Color Non-Waterproof DC5V 16.4ft that will be used.

So, just having the strips spaced out enough from each other to fit under each of the squares and that the strip length matches with the board's dimensions was all that is necessary to assemble the matrix. Then, to power the matrix it just needed to be connected to the LED drivers and as mentioned before two of the ALITOVE 5V 10A AC to DC Power Supply Adapter Converter Charger 5.5x2.1mm Plug for WS2811 2801 WS2812B LED Strip Pixel Light was needed to power the LED strips without burning them out.

Then, finally, they were connected to the Arduino Mega 2560 to be able to program each individual LED. Thankfully, it was not too complicated to set up and it's straightforward. The LED's were all connected in a sequence and this is represented as follows within Figure 23.

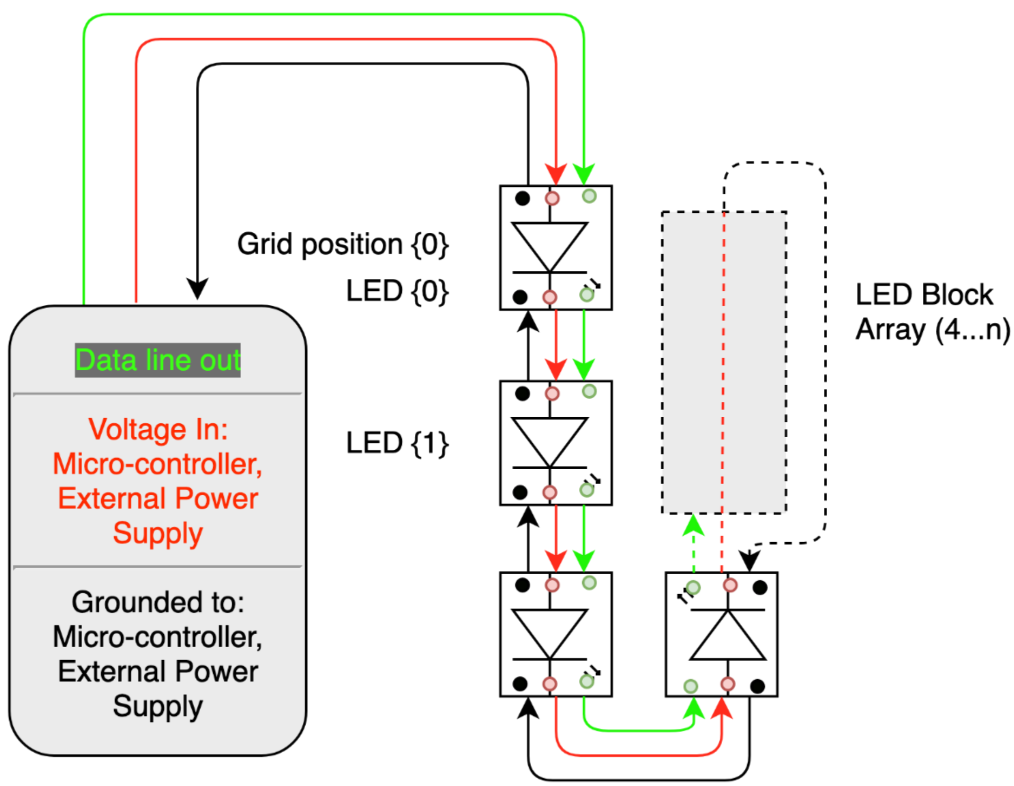


*Figure 23: WS2812B LED Sequence*

## 8.2 Hardware Test Plan

After connecting the system with the Arduino MEGA 2560 microcontroller, the most important aspect of the project that needed to be tested was the power supply of the stepper motors. It was crucial to observe how much current was the power supply drawing to the system when running a few scenarios of gameplay and mitigate the risk of being too hot on the stepper motors and the system itself.

The LEDs were going to be tested by building small prototype on the breadboard and using the Arduino MEGA 2560 again and connecting the LEDs, and LED drivers in order to observe if the microcontroller can change the brightness of the LEDs.

A ‘game time’ run was initiated and do multiple scenarios when LEDs took part. In a case when the ‘players’ were getting ambushed or attacked by enemies, a reddish color appeared by the LEDs and was controlled by the Dungeon Master. The ‘Dungeon Master’ was using an app for controlling the brightness and colors for the LEDs to perform.

If the LEDs didn’t respond the commands from the application, then the connection between the Arduino MEGA 2560 and the LEDs had to be checked. The Reed switches were to be tested as well. The electromagnet in the XY plotter used evasive maneuver to avoid certain Reed switches when holding magnetic game piece to the desired location.

Since the Reed switches did not require external source power, they were tested individually when the XY plotter started to move the game piece around. Another consideration was to avoid the Reed switches not to be so close to the LEDs during gameplay. They both needed to work independently.

The ideal location of putting on these Reed switches were under the gameboard; however, the LEDs were also going to be there. Several test runs took place and changed locations of the LEDs and the Reed switches and then determined which ideal assembly took part of this project afterwards. Table 47 showcases the procedures that the group followed.

**Table 47: Testing Procedure**

| | |
|---|---|
| **1** | Checked the functionality of the Stepper Motors and the Chopper Driver Controller for power supply |
| **2** | Checked the connection between the Arduino MEGA 2560 and the LEDs |
| **3** | Ran a few scenarios that require LED colors by the Dungeon Master |
| **4** | Checked for XY plotter’s movement for the game piece |
| **5** | Checked the functionality of the Reed Switches on the gameboard |
| **6** | Ran a few scenarios when the game piece was required to move from one location to another location without any interference of the Reed Switches |

### 8.2.1 Sensor Plan

For the Reed-Switch sensors, they are going to be tested for its strength and its functions for the project. In this case, the Reed-Switch sensor will be built on the breadboard and test the its strength with a game figure on the display. The game figures have a small mass and applying a magnetic piece on the bottom to test these types of sensors. The goal for this is to see how effective these sensors can respond with different types of Reed-Switch sensors.

To determine which product can be used in the project was to test the distance between the game figure and the sensor. The shorter the sensing distance, better results. The game figures were affected on a larger distance as they were meant to be put in place.

For the Reed-Switch sensors, they were tested for their strength and their functions for the project. In this case, the Reed-Switch sensor was tested on the breadboard and tested its strength with a neodymium magnet on the display. The game figures have a small mass and applied a neodymium magnet on the bottom to test these types of sensors.

The goal for this was to see how effective these neodymium magnets responded with different types of Reed-Switch sensors. To determine which product was used in the project was to test the distance between the game figure and the sensor. The shorter the sensing distance, better results. The game figures were affected on a larger distance as they were meant to be put in a short distance.

### 8.2.2 LED Plan

For the LEDs, they were connected with the Arduino MEGA 2560 microcontroller. From there, the PWM signal was used in this process to make sure that the LEDs had certain brightness levels for gameplay. In this case, the LEDs needed to have passed certain requirements for the project. To do that, the LED product chosen for the project needed to be tested before going large-scale. The best method was to put the circuit design of each distinct LED product all on the breadboard.

Also, the power system was crucial for the connection of LED drivers which then determined which number of LEDs were used on the 10 x 13 grid gaming surface. The distribution of power was critical for those LEDs by verifying how much current was each LED able to manage. To avoid any overheating and damage to the LEDs. On a breadboard, an LED driver was connected to a set of LEDs and was able to troubleshoot any connection errors.

### 8.2.3 XY Plotter Plan

The XY plotter was bought assembled with a working area of 10 X 13 and did slight configurations on the XY plotter which included implanting an electromagnet, motor controllers and changing the stepper motors as well. The XY plotter. As soon as these configurations were finished, the placement of the Reed switches and LEDs took place.

Each LED was placed on each square coordinate of the grid along with a Reed Switch. The connectivity for these components was crucial during this process for every square coordinate. Testing took place for the stepper motors with the connection for the motor controller and the Arduino MEGA 2560.

They ran a few gameplay scenarios when moving the game piece and purposely made a few errors do that the motor controller detected those errors for its feedback functionality. That was the first stage of the plan was to make sure that the stepper motors, motor controller, and Arduino MEGA 2560 were satisfied.

The next stage was testing those components with the interference of the Reed Switches and the LEDs. Those LEDs will be connected to the LM3402 and the LM3402 was connected to the one of the pins of the Arduino MEGA 2560. From there, few gameplay scenarios were initiated and confirmed that the LEDs and the Reed switches were functional.

Also, the electromagnet will be configured in such a way so that the Reed switches avoided interference with each other during gameplay. This was all summarized in within Table 48.

**Table 48: Test Plan for the XY Plotter**

| | |
|---|---|
| **Step 1** | – Connect the motor controller, stepper motors with the Arduino MEGA 2560<br>– configure the magnetic holder to the right specifications |
| **Step 2** | – Connect the LEDs, and the Reed Switches on the 16x16 grid space with the LM3402 and LED drivers |
| **Step 3** | – Computer engineering students will test the XY plotter functionality with multiple scenarios and make finishing touches. |

By following these main steps for the project, the specifications and standards were met. However, the discrepancies of the project arise such as coding malfunction, overheated and damaged stepper motors, and lack of connection between the microcontroller Arduino MEGA 2560.

## 8.3 Software Test Setup

When it came to Software test set up, the group needed to check for potential bugs before it was made live or moved into the production environment. During this stage many issues occurred such as software security, the functioning of the site, its access to users and its ability to handle traffic.

The following testing types may be formed by using depending of the testing requirements.

### 8.3.1 Functionality Testing

This test set up was used to check if their product worked as per the specifications intended as well as the functional requirements the group designed in the developmental documentation. Functions were tested by feeding them input and then inspecting the output. The coding framework was tested by running sample events and verifying that the output meets the project's expectations. The number of sample events had to be higher.

This type of tests ensured their test setup was done right by checking that all the requirements were properly satisfied by the application.

## 8.4 Software Test Plan

The software test plan was set into place to make sure that the web application and tabletop software development procedure was done efficiently and to minimize debugging sessions. Errors that occurred along the way should be contained within their functionalities.

### 8.4.1 U.I.S.A Testing

U.I.S.A Testing referred to the level of software testing the team did in their project. The term U.I.S.A stands for Unit testing, Integration testing, Software (System) testing and Acceptance testing. In order to perform a high-quality testing plan for their project, the group needed to follow each software testing in a specific format as showed in Figure 24 below.

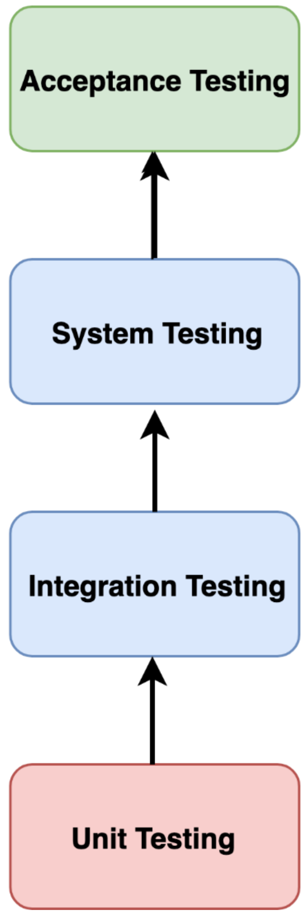


*Figure 24: Software Level Testing*

The first level of software testing the group needed to proceed as explained in the graph above was Unit testing. The purpose of Unit testing for their project was to start off testing small pieces of code written, mainly individual functions or components. Once the group picked a function to test, it gave some inputs and then checked for the function output. By doing so, it validated that each unit of the software performed as expected.

Proceeding with Unit testing had some benefits in the long term such as increasing their confidence in changing and maintaining code, If the group implemented a good unit test where it ran every time their code was edited this allowed them to promptly catch any errors introduced due to the changes. Their code also needed to be modular in order to make unit testing possible, this allowed their code to be more reusable. A few popular tools the group considered using are Jasmine, Mocha and tape.

After setting up their Unit test, the next software test needed was the integration test. The purpose of this test was to expose faults in the interaction between integrated units. The group used the units previously tested and now combined them and tested them as a group. After doing some research, the group considered implementing test drivers and test stubs to assist the team members uring integration since they were effective and saved time.

The group also implemented two types of integration tests which are component and system. Component integration tests were used to expose errors in the interaction and interface between integrated components and System integration tests that were excellent for testing systems and packages as well as interfaces to external organizations.

Once the integration test was finished, the group created a proper detail design document where it defined interactions between each unit. This was important so that anyone working on the code was able to see what functions were tested as well as the interaction between units that were defined.

The second to last test needed was the System testing, in this section the group tested the completed and integrated software. The purpose of this test was to evaluate the system's compliance with their specified requirements. One method the group implemented is called the black box testing, in which the internal design/implementation/structure of the item were testing is not known to the tester.

This method was great in finding incorrect or missing functions as well as interface errors, behavior or performance errors, initialization and termination errors. The only problem the group found for this method was that only a small number of possible inputs can be tested, this left many program paths untested.

The last phase of the software testing process the group implemented is called Acceptance testing, this test was done to evaluate their system's compliance with the business requirements and determine whether a finished product is ready acceptable for delivery. Then the customer either accepted or rejected the product developed.

For their project, the type of Acceptance test the group conducted is called operational Acceptance Testing (OAT). This was done to determine the operational readiness of their product. This includes testing of compatibility, recovery, technical support availability, maintainability, etc. This was all done to assure the stability of the product for their presentation and any potential customer who was interested on it.

Applying U.I.S.A had also some potential costs or setbacks, the most noticeably the group found was that creating a high-quality acceptance tests required significant effort and a good amount of time spent on it. Also, if the group encountered a “fragile” test issue, other teams have explained it can become burdensome for the maintenance part.

Although setting up these phases of software level tests seemed like it will take a good amount of the given time available. The group had determined it was time necessary spent since they wanted to make sure the product that was presented had no software connections errors.

### 8.4.2 Code Quality

Code quality defined code that was determined good (high quality) as well as code that was bad (low quality). This was all subjective as different teams used different definitions, for example for an automotive developer good code quality may mean something different compared to a web application developer.

Code quality was important for the overall software quality, which impacted how secure, safe and reliable the codebase is. Today, high quality is critical for many teams and managers. Hence why their group focused on some attributes to make sure the code written for their project was high quality.

When their code was being written, the first thing the group looked for when reviewing was the clarity of it, the team needed to ensure that their code was easy to read and oversee for anyone who had not created the code. The group needed to make sure it was easy to understand and therefore maintain and extend the code.

This was important for group projects since in their case, there were mainly two people working on the code for their project. This was of great importance for either one of them to make changes. Once the code was clear, the group then added comments to the code to explain its role and functions.

Once the code had been checked for clarity and comments were added, their next step was to check for any bugs, the less bugs the code had meant the higher its quality was. These requirements were a priority that their code needed to follow. Their plan was to meet all the previously mentioned attributes to ensure the code that was written is high quality.

Besides following these requirements, the group had to set up additional testing plan methods for code quality. Their first step was to set up a version control tool, after doing some research the group determined that GIT would be the preferred control tool. The reason being it provided a system that was easy to track which separated live product from less stable branches with unpublished features.

Once the feature was finished, a pull request was sent to GitHub. This appeared in a section that was ready to review waiting for a project member to review. Once it was reviewed by a team member and if it met the requirements needed, it was then merged to a development branch.

This was a great system when it came to controlling versions and showed every person’s code that had worked on it. Once the version control had been set up, their next step was adding functional tests. Functional quality tests demonstrated if the code was working fine or not. A pyramid-shape diagram shows the test process and where the group should be placing their efforts, the team should spend the most time doing unit tests, lesser integration tests and even lesser End-to-end tests as explained in the graph below in Figure 25.

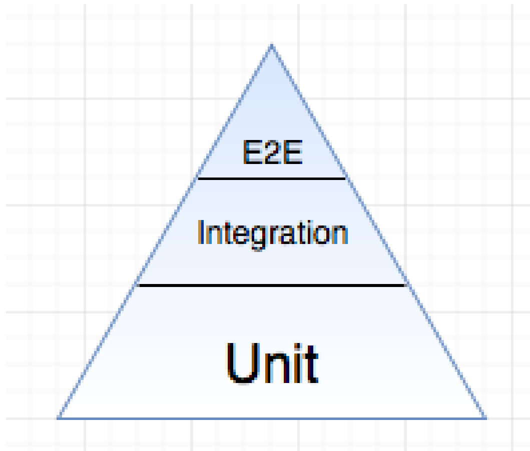


*Figure 25: E2E/Integration/Unit Pyramid*

### 8.4.3 Error Handling

The main objective of the Error Handling technique was the anticipation, detection and resolution of applications, programming and communications errors. There were specialized programs which are called error handlers that are available, and that the team used. The best programs found forestall errors. If possible, recovered from then without closing/terminating the application. If the file failed, it terminated an affected application and saved the error information to a log file.

For their project, the error handling testing was done by the computer engineer students that worked on it, the most experienced person was the lead since he has full knowledge of the code and database.

Their approach was to generate improper transactions in-between the proper transactions and then check how the system behaved during the improper transaction and identified any given problem. Also, by using the improper master data and then checking the application whether it can identify the problem and informed the users.

# 9.0 Project Integration

The integration of all our components on the project which using some parts of the Arduino MEGA 2560 and the microcontroller ATMEGA 16U along with the connection of the reed-switch matrix with a built encasing, the connection of the electromagnet and the XY plotter.

## 9.1 Hardware Integration

This section goes over the PCB integration and the building process of the enclosure, and physical aspects of our LED and reed switch matrix. The creation of these elements was integral in proceeding forward with consolidating the different elements of feature set. This provided a fast prototype test-bed to rapidly adjust the intertwined elements of the system.

### 9.1.1 Project Enclosure

The project enclosure was necessary to encase the layers that the project needed to function. The enclosure from bottom layer to top layer included the XY plotter, LED matrix, reed switch matrix, and then the gameboard all the way on top. The materials used to make the enclosure included fiberboard, wood sticks, gorilla glue for wood, wooden borders (for decoration), black spray paint, and frost spray paint. The way the enclosure was measured to be built was by measuring the XY plotter's length, width, and height. The XY plotter was the foundation to build the enclosure since the XY plotter is what controls everything else that makes the project.

The XY plotter was the main aspect of our project to determine the size of the enclosure and acrylic boards that would be used for the layers. It was easy to measure the width and length of the enclosure because that was very straightforward from the XY plotter. It was a bit of a challenge to measure the height of the controller because the distances of the LED matrix layer, reed switch matrix layer, and electromagnet spacing had to be taken into consideration. It was not that much of an issue though because the distance between the game piece magnet and XY plotter electromagnet was already measured beforehand, and the LED matrix, reed switch matrix was also built.

Having those measurements then made it straightforward to determine the height of the enclosure. After getting the measurements the fiberboard was cut with an electric saw, then the pieces were sanded down with sandpaper to even the boards out, after that, the boards were gorilla glued together and left to dry out overnight. The wood sticks that were bought were also glued on the inside of the board to be able to hold the LED, reed switch matrix layer, and the very top gameboard layer. The wooden borders were then glued on the top edges for decoration. After the enclosure's glue dried up and it was tested to be sturdy, then it was spray-painted in black. The wood sticks were measured and put to be able to have the LED matrix, reed switch matrix, and top gameboard layer to just slide in. The very top gameboard acrylic board was spray-painted in frost to be opaque.

### 9.1.2 PCB implementation

The PCB schematic consisted of using the important components such as ATMEGA 2560 surface mount electronic chip, ESP8266-12E Wi-Fi module, A4988 stepper drivers which is connected to NEMA 17 stepper motors. The shift registers, which comprised of two 74HC165's and 74HC595's and the connection of the TIP120 transistor with the electromagnet. Most of the passive elements such as ceramic capacitors of 0.1uF and 47uF, and resistor arrays valued of 10Kohm and 2Kohm, are maintained onto the schematic design.

To avoid overheating, the heatsink was considered for the power regulator where all electronic operations are taking place. However, the heatsink was not placed since the duration of the demo was roughly 10 minutes. An CPU fan was contemplated for additional regulation of the operating temperature of the PCB board but was discarded in the final assembly. Two power jacks are placed on the corner of the PCB board. The first power jack is connected to the ATMEGA 2560 chip, the Wi-Fi module, and the shift registers with 5V. The second one is to power 12 volts to the A4988 stepper drivers connected to the stepper motors. A third power jack was connected to a screw-terminal adapter, which in turn was connected to the electromagnet configuration.

## 9.2 Software Integration

This section goes over the software implementation on the webserver, Wi-Fi-communication, and the SPI communication between the Master Atmega2560 Controller and our Slave ESP8266. The server-client communication is key to for multiple features to run correctly and is the core feature to synchronize all the connected platforms. A functional hardware prototype provided rapid development towards creating the software on the different embedded systems and web app client and server.

### 9.2.1 Wi-Fi Communication

The chosen Wi-Fi communication between the board and microcontroller was the ESP8266-12e module. The reason the group chose this model for our project was because it is reliable in stablishing wireless network connections. It included an integrated cache which improved the performance of the microcontroller without utilizing too much memory. This allowed communication between the front end of our application and the microcontroller which in returned sent commands and action to the board.

### 9.2.2 Communication Protocol

Socket-IO was the initial library by the group for bi-directional communication between the server and clients. It had more features than the original WebSocket protocol chosen at the beginning of the project. The group was able to establish a chat system using this protocol that communicated between users in the game and also showed other players who was online, who disconnected and how many players were connected.

The communication protocol that was established between the Node JS server and the ESP8266 was the WebSocket's protocol. From there we were able to send JSON string packages from the webserver to the Wi-Fi module, and deserialize the string data into a proper JSON object. We can discern what type of command is established and then process the rest of the data from the resulting type. This is reflected in our move command and another command that loads the server matrix obstacle data onto the physical 2D grid. These obstacles are shown as lit red LED's on the physical tabletop.

### 9.2.3 SPI Protocol Integration

When a data package arrives onto the ESP8266 Wi-Fi module and is processed as a JSON object, the resulting data needs to be translated over to the Atmega2560 controller. As our communication protocol between the two devices, we used SPI as planned, though there was a limitation that came with using this method. The SPI libraries within the ESP8266 Arduino Core contained a 32-byte limit on individual packages, so to circumvent this we split apart data arrays and sent multiple packages through the SPI bus.

The controller was considered the master device, so it was responsible on determining the type of message and asking for the right amount of data to be processed. Certain commands are static such as the grid matrix data but move actions can contain different lengths of data. This meant that the master device has verify the data length and proceed from there.

### 9.2.4 Webserver and Data storage

NodeJS was the chosen webserver by the group utilized in the web application. It allowed the group to create high performance applications within our project and it only required one programming language which was JavaScript. Many other webservers offered great features but due to the lack of time to learn another programming language such as PHP or phyton the group felt this was the ideal webserver for the project.

# 10.0 Project Management

Since the project consisted of four team members, it was important to manage tasks, scheduling, and distribute work effectively. The most important aspect of organization and management is scheduling. In this case, team members had to be aware of their availability to dedicate their contribution to the project. Frequent meetings were necessary to bring new ideas and solutions to mitigate risks in the future.

## 10.1 Version Control

To make sure that their project development is a smooth process, the group used a version control tool to distribute and manage their development pipeline. There are two different applications that needed to be developed, which is the Web application for the removed users from the main group, and the controller application that monitors, controls and receives all changes that happened on the board and the database. Every iteration of various features needed to be thoroughly bug tested and go through quality assurance to meet their defined feature set.

There are a variety of version control applications that were used but the team needed the right application to meet their requirements to manage the project, as well as integrated nicely with their development tools and server environment.

### 10.1.1Github

GitHub is an open-source software management when it comes to coding development. Its features were very effective when building and testing the codes seamlessly. Computer engineering students were going to be communicating not only to themselves, but to other experienced coders that provided essential help. Any lines of code built by an experienced coder; this coder was referenced to their contribution in the final documentation.

The most effective feature that was used in GitHub is the ability to add branches under a main branch called master. Unlike other centralized control system. GitHub branches were cheap and easy to merge, they provided the group an isolated environment for every change to the codebase. When the project was started, the computer engineer students created a new branch which ensured the master branch always contained production-quality code.

### 10.1.2Bitbucket

Bitbucket is another source like GitHub that was used to get source codes. It is used more for professional developers, but it’s another website that was useful for the computer engineers in the group to find and reference codes that was used to program the Arduino Mega 2560. Of course, any code that was referenced was given credit to appropriately.

This software also has a free plan for groups up to five people and includes Jira software integration. It fits their group size and combines a task tracking system that helped with expediting their development and consolidated communication for across the project versioning and tasks. Jira is explained further on within section

10.2.2. The feature set allows for all functionalities of using git, and different branches can be made on top of the master branch, such as the staging, dev, and hotfix branch.

The dev branch is where new features were “branched” out, finished, and merged back within the branch. This is where most of work on the software side was done, and staging branch was where final integration testing was performed before pushing the master branch for production use.

Three projects were created for their product. One being for the web application, another for the server and database communication/modeling, and the last for the Tabletop application. These separate the major tasks into these distinct categories so that the team focused on one or the other without creating conflicts on other systems that weren’t related.

## 10.2 Task management

This group consisted of two computer engineering and the other two were electrical engineering students. Therefore, the two electrical engineering students were responsible of implementing an XY plotter, connectivity of the LEDs, Reed-Switches, and the Stepper motors, and the Arduino MEGA 2560. The two other computer engineering students were responsible for web and application development. However, this type of task distribution was changed ever so slightly due to limited time.

**Table 49: Task Distribution by Each Team Member**

| | |
|---|---|
| **Team Member # 1 (CpE)** | Developing a User Interface for Web application/Participate in using |
| **Team Member # 2 (CpE)** | U.I.S.A Testing / Web and application development |
| **Team Member # 3 (EEE)** | Configuration of the Magnetic Holder/Connection to the Stepper Motors and Motor Controller |
| **Team Member # 4 (EEE)** | Placement of LEDs, Reed Switches on the 10x13 grid connecting from that grid to the XY Plotter |

Depending on the code difficulty and the type of code used in this project, electrical engineering students were able to learn and help the computer engineering students if the code lines were too overwhelming. This can be applied to the computer engineering students; they contributed their help if the two electrical engineering students had difficulty. This process was shown in Table 49.

This was the ideal distribution for all the students in their respective majors. However, this distribution table was flexible to change if one of the team members did not accomplish their own individual tasks.

The group main task was setting up reminders and assignments each week in order to accomplish all requirements set from the beginning and a fully working project in the end of the semester.

### 10.2.1Google Tasks

In the google calendar, there is a feature that produced tasks for each team member, and they were able to evenly distribute it during the week. In the task section, each team member specified the content and estimated the page count for that content. As soon as that team member finished writing those pages, they marked that task “done” so that all the team member was able to see progression on the page count.

From there, they were able to police themselves and motivate others to write the page count needed for the project. This was a very flexible procedure since the team members contributed sources to others to finish their content and supported them as much as they can.

### 10.2.2JIRA

Another collaborative online platform, however, this online collaboration has multiple features in regards of software development. This online platform created consistent planning and had a visualization of the team members’ progress when delivering codes. This can be beneficial to the team members that have a computer engineering background when sharing codes to each other and improvise their lines of codes in order to design a web application and interface.

The software also has benefits for tracking the hardware changes that needed to be implemented. The way that Jira works is that the user can separate the main objectives into multiple different projects. One project would be for the web application and one for the physical tabletop project. The users would both have to integrate with each other through the internet, so when it came to any parts that need the web component, an integration project can be utilized for the communication.

Once these projects are made, then Jira tickets were generated by their team members on various tasks and set prioritizations and order for all the features that needed to be made. The major requirements for their project were at a high prioritization, while smaller features such as graphics improvements on the web application, were at the lowest.

### 10.2.3Microsoft SharePoint

Microsoft SharePoint is a collaborative online platform so that the team members were able to view their progression in terms of page count and what type of information can they contribute. After the writing their respective content done, they would notify other team members about their completion and give support to those in need in order to complete the project report. In this platform a folder was created so that they were able to link multiple source documents that can fulfill a great understanding that was necessary for the project. Since SharePoint is part of Office 365, it became the nexus point on where it was branched out to create all the documentation, graphs, and data sheets. Due to having an account through UCF, the group ended up using Office 365’s resources to its utmost potential in order to smooth out the processes of writing out documentation and increasing the amount of time towards actual development.

The use of the Arduino MEGA 2560 was not only simple, but rather required reading multiple documents and resources regarding of this microcontroller and learned the connectivity to the overall system of the project which is the XY plotter with the stepper motors and the motor controller. Essential coding references were shared as well among the group so that they were able to learn to code and modify as much as they can to satisfy the requirements of the microcontroller and the implication of web and user application.

Also, it has feature of having a calendar that contains many functions which was helpful for group communication. It was also integrating task tracking which was auto-emailed their progress on various documents that needed more progress. The calendar contained individual tasks that each team member needed to accomplish and sent an email notification if the task was not completed after the deadline. If the share-cloud server was being used, the password was created for team members for security reasons.

## 10.3 Scheduling

Scheduling is the most important step for a group-collaborative project. One team member was assigned to schedule either online meetings using a Discord platform, Zoom, or attending by person. Online meetings were frequently used due to distant homes or time constrained by work. Discord and Zoom meetings would be used regularly to discuss the project.

The main aspect of this process was the flexibility. If a team member was not able to attend the meeting due to family or work, the meeting continues, however, one of the team members notified the absentee the contents of the meeting or sent an email that summarized the meeting during that time. This approach was very effective.

### 10.3.1 Google Calendar

By using the Google Calendar, it provided a visualization of the working times and class times that each of the team members had. That way, the meetings were made based off the google calendar. This a great tool for task management and providing a “check-list” feature for page count distribution. Each week, the page count would be roughly 6-7 pages which then the google calendar alerted the team member to write their own page count distribution in every week.

The Google Calendar have features for the team members to post notes regarding on the specific work for the project. They posted notes to aid and give feedback when the content was hard to comprehend. They interacted by creating attached meetings and applied on certain days for all the team members to know.

## 11.0 Works Cited


[1] Team11, "Ghost Chess," [Online]. Available: https://github.com/2083008/GhostChess#required_hardware.

[2] Amazon , "KAPATA Digital RGB Strip Individually Addressable LED Strip WS2812B WA2812 5m 30IC-30LED/M White PCB Waterproof Dream Color DC5V," [Online]. Available: https://www.amazon.com/Individually-Addressable-Digital-30IC-30LED-Waterproof/dp/B014QKWJDU/ref=asc_df_B014QKWJDU/?tag=hyprod-20&linkCode=df0&hvadid=232831470222&hvpos=1o1&hvnetw=g&hvrand=177017695530723040 49&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvloc.

[3] Unknown, "Latching and Non-Latching Hall Effect Sensors," 29 January 2014. [Online]. Available: arduinogizmo.blogspot.com/2014/01/latching-and-non-latching-hall-effect.html.

[4] Mouser Electronics, "MAX16802B Datasheet," [Online]. Available: http://html.alldatasheet.com/html-pdf/338070/MAXIM/MAX16802B/1258/8/MAX16802B.html.

[5] [Online]. Available: https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html.

[6] "Eletronic Tutorials: Hall Effect Sensor," [Online]. Available: https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html.

[7] "What Is Amazon EC2?," Amazon, 2019. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html.

[8] R. Santos, "Guide for WS2812B Addressable RGB LED Strip with Arduino," Random Nerd Tutorials, [Online]. Available: https://randomnerdtutorials.com/guide-for-ws2812b-addressable-rgb-led-strip-with-arduino/.

[9] unknown, "Analog/Digital Conversion with Microcontrollers," 17th April 2006. [Online]. Available: https://www.bipom.com/documents/lectures/Analog%20To%20Digital%20Conversion%20with%20Microcontrollers.pdf. [Accessed 2019].

[10] "Coding conventions," 28 July 2019. [Online]. Available: https://en.wikipedia.org/wiki/Coding_conventions.

# 12.0 Appendix A

**From:** Oguz Murtezaoglu <oguz@bipom.com>

**Subject: Re: Requesting permission to use materials relation to senior design capstone project**

**Date:** July 29, 2019 at 7:19:43 AM EDT

**To:** Hubert Barrantes <hubert.barrantes@Knights.ucf.edu>

**Reply-To:** Oguz Murtezaoglu <oguz@bipom.com>

Sure, no problem. Good luck with your project !

Best Regards

Oguz "Oz" Murtezaoglu

BiPOM Electronics, Inc.
9788 Clarewood Dr. 306
Houston, Texas 77036
Phone: 713-283-9970
Web: https://nam02.safelinks.protection.outlook.com/?url=www.bipom.com&da

ta=02%7C01%7Chubert.barrantes%40Knights.ucf.edu%7C96230bcd01664291321608d71416a949%7C5b16e18278b3412c919668342689eeb7%7C0%7C1%7C636999959874823817&sdata=xTTMTPUMqMpBhZvUb%2F567wkV01ts6Yo%2BxfjdepCz7ls%3D&reserved=0



On 7/28/2019 3:17 PM, Hubert Barrantes wrote:

To whom it may concern.

I would like to use a graph on your Analog/Digital Conversion with micro controllers document if possible for my capstone project. Is it ok to use the diagram below for documentation purposes?Below is the link
Please let me know.
Hubert Barrantes

https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.bipom.com%2Fdocuments%2Flectures%2FAnalog%2520To%2520Digital%2520Conversion%2520with%2520Microcontrollers.pdf&data=02%7C01%7Chubert.barrantes%40Knights.ucf.edu%7C96230bcd01664291321608d71416a949%7C5b16e18278b3412c919668342689eeb7%7C0%7C1%7C636999959874823817&sdata=SMGprugwo%2FyW2ufLM%2BF55tNLcN5DZQ50wDPiohO0UBI%3D&reserved=0